X1000

IoT Application Processor

Programming Manual

Release Date: Jan. 13, 2016

北京君正集成电路股份有限公司

Ingenic Semiconductor Co.,Ltd.

X1000 IoT Application Processor

Programming Manual

Disclaimer

Ingenic Semiconductor Co., Ltd.

Ingenic Headquarters, East Bldg. 14, Courtyard #10,
XIbeiwang East Road, Haidian District, Beijing 100193, China,
Tel: 86-10-56345000
Fax: 86-10-56345001
Http: //www.ingenic.com

CONTENTS

CONTENTS i

TABLES i

FIGURES i

Section 1 Overview

1 Overview 2

1.1 Block Diagram 2
1.2 Features 2
1.2.1 CPU Core 2
1.2.2 Image Core 3
1.2.3 Display/Camera/Audio 3
1.2.4 Memory Interface 5
1.2.5 System Functions 6
1.2.6 Peripherals 7
1.2.7 Bootrom 9

Section 2 Core Functions

2 CPU 11

2.1 Block Diagram 11
2.2 Extra Features of the CPU core 13
2.3 Instruction Cycles 13
2.4 PMON 16
2.4.1 Fundamental 17
2.5 Partial Kernel Mode 17

Section 3 Image Core

3 JPEG 19

3.1 Overview 19
3.2 Register Definition 19
3.2.1 Task trigger (TRIG) 19
3.2.2 VDMA status (STAT) 20
3.2.3 Global control information 20
3.2.4 JPGC trigger 21
3.2.5 JPGC Status 22
3.2.6 Bitstream buffer address 22

3.2.7 Component 0 plane buffer address 23
3.2.8 Component 1 plane buffer address 23
3.2.9 Component 2 plane buffer address 24
3.2.10 Component 3 plane buffer address 24
3.2.11 MCU number 25
3.2.12 Re-Sync-Marker gap number 25
3.2.13 Component configure information 25
3.2.14 EFE control 26
3.2.15 Encoder slice geometry information 26
3.2.16 RAW plane source buffer address 27
3.2.17 RAW destination buffer basement address 27
3.2.18 RAW plane stride 28
3.3 Table definition 28
3.3.1 Huffman Base Table 28
3.3.2 Huffman MIN Table 28
3.3.3 Quantization Table 28
3.3.4 Huffman Symbol Table 29
3.3.5 Huffman ENC Table 29

Section 4 Display/Camera/Audio

4 SLCD Controller 31
4.1 Overview 31
4.2 Pin Description 31
4.3 Block Diagram 32
4.4 Register Description 32
4.4.1 Configure Register (LCDCFG) 35
4.4.2 Control Register (LCDCTRL) 35
4.4.3 Status Register (LCDSTATE) 36
4.4.4 OSD Control Register (LCDOSDCTRL) 37
4.4.5 Background0 Color Register (LCDBGC0) 37
4.4.6 Display Area Horizontal Start/End Point (LCDDAH) 38
4.4.7 Display Area Vertical Start/End Point (LCDDAV) 38
4.4.8 Foreground 0 XY Position Register (LCDXYP0) 38
4.4.9 Foreground 0 Size Register (LCDSIZE0) 39
4.4.10 REV Signal Setting (LCDREV) 39
4.4.11 Interrupt ID Register (LCDIID) 39
4.4.12 Descriptor Address Registers (LCDDAx) 40
4.4.13 Source Address Registers (LCDSA) 41
4.4.14 Frame ID Registers (LCDFIDx) 42
4.4.15 DMA Command Registers (LCDCMDx) 42
4.4.16 DMA OFFSIZE Registers (LCDOFFSx) 43
4.4.17 DMA Page Width Registers (LCDPWx) 43

4.4.18 DMA Commend Counter Registers (LCDCNUMx) 44
4.4.19 DMA Commend Counter Registers (LCDCPOSx) 44
4.4.20 Foreground x Size in Descriptor (LCDDESSIZEx) 45
4.4.21 Priority level threshold configure Register (LCDPCFG) 45
4.4.22 LCD Arbiter Priority(LCD_PCFG_ARB) 46
4.4.23 SLCD Configure Register (MCFG) 47
4.4.24 SLCD Configure New Register (MCFG_NEW) 48
4.4.25 SLCD Control Register (MCTRL) 51
4.4.26 SLCD Status Register (MSTATE) 52
4.4.27 SLCD Data Register (MDATA) 52
4.4.28 SLCD Wait Time Register (WTIME) 53
4.4.29 Address Setup and Hold Time Register (TASH) 53
4.4.30 Slow Mode Wait Time Register(SMWT) 53
4.5 SLCD Controller Operation 54
4.5.1 Set SLCD Controller AHB Clock and Pixel Clock 54
4.5.2 Enabling the Controller 54
4.5.3 Disabling the Controller 54
4.5.4 Resetting the Controller 55
4.5.5 Frame Buffer 55
4.6 System Memory Format 55
4.6.1 Data format 55
4.6.2 Command Format 57
4.7 Transfer Mode 58
4.7.1 DMA Transfer Mode 58
4.7.2 Register Transfer Mode 59
4.8 Timing 59
4.8.1 Parallel Timing 59
4.8.2 Serial Timing 60
4.9 Operation Guide 60
4.9.1 DMA Operation 60
4.9.2 Register Operation 61
5 Camera Interface Module 62
5.1 Overview 62
5.1.1 Features 62
5.1.2 Pin Description 62
5.2 CIM Special Register 62
5.2.1 CIM Configuration Register (CIMCFG) 63
5.2.2 CIM Control Register (CIMCR) 66
5.2.3 CIM Control Register 2 (CIMCR2) 67
5.2.4 CIM Status Register (CIMST) 68
5.2.5 CIM Interrupt Mask Register (CIMIMR) 69
5.2.6 CIM Interrupt ID Register (CIMIID) 70

5.2.7 CIM Descriptor Address (CIMDA) 71
5.2.8 CIM Frame buffer Address Register (CIMFA) 71
5.2.9 CIM Frame ID Register (CIMFID) 71
5.2.10 CIM DMA Command Register (CIMCMD) 72
5.2.11 CIM Window Size (CIMWSIZE) 73
5.2.12 CIM Window Offset (CIMWOFFSET) 73
5.2.13 CIM Frame Size Register (CIMFS) 73
5.3 CIM Data Sample Modes 74
5.3.1 Gated Clock Mode 74
5.3.2 ITU656 Interlace Mode 75
5.3.3 ITU656 Progressive Mode 76
5.4 DMA Descriptors 76
5.5 Software Operation 77
5.5.1 Enable CIM with DMA 77
5.5.2 Operations for RXFIFO Overflow 77
5.5.3 Operations for Frame Size Error 77
6 Audio Interface Controller 79
6.1 Overview 79
6.1.1 Block Diagram 80
6.1.2 Features 80
6.1.3 Interface Diagram 81
6.1.4 Signal Descriptions 82
6.2 Register Descriptions 83
6.2.1 AIC Configuration Register (AICFR) 84
6.2.2 AIC Common Control Register (AICCR) 86
6.2.3 AIC I2S/MSB-justified Control Register (I2SCR) 89
6.2.4 AIC Controller FIFO Status Register (AICSR) 90
6.2.5 AIC I2S/MSB-justified Status Register (I2SSR) 92
6.2.6 AIC I2S/MSB-justified Clock Divider Register (I2SDIV) 93
6.2.7 AIC FIFO Data Port Register (AICDR) 93
6.2.8 SPDIF Enable Register (SPENA) 94
6.2.9 SPDIF Control Register (SPCTRL) 94
6.2.10 SPDIF State Register (SPSTATE) 95
6.2.11 SPDIF Configure 1 Register (SPCFG1) 96
6.2.12 SPDIF Configure 2 Register (SPCFG2) 97
6.2.13 SPDIF FIFO Register (SPFIFO) 98
6.3 Serial Interface Protocol 99
6.3.1 I2S and MSB-justified serial audio format 99
6.3.2 Audio sample data placement in SDATA_IN/SDATA_OUT 101
6.3.3 SPDIF Protocol 102
6.4 I2S Operation 103
6.4.1 Initialization 103

6.4.2 External CODEC Registers Access Operation 104
6.4.3 Audio Replay 104
6.4.4 Audio Record 105
6.4.5 FIFOs operation 106
6.4.6 Data Flow Control 107
6.4.7 Audio Samples format 108
6.4.8 Serial Audio Clocks and Sampling Frequencies 110
6.4.9 Interrupts 114
6.5 SPDIF Guide 115
6.5.1 Set SPDIF clock frequency 115
6.5.2 PCM audio mode operation (Reference IEC60958) 115
6.5.3 Non-PCM mode operation (Reference IEC61937) 115
6.5.4 Disable operation 116
7 PCM Interface 117
7.1 Overview 117
7.2 Pin Description 117
7.3 Block Diagram 118
7.4 Registers 118
7.4.1 Registers Memory Map 119
7.4.2 Register Description 119
7.5 PCM Interface Timing 125
7.5.1 Short Frame SYN 125
7.5.2 Long Frame SYN 126
7.5.3 Multi-Slot Operation 126
7.6 PCM Operation 127
7.6.1 PCM Initialization 127
7.6.2 Audio Replay 127
7.6.3 Audio Record 128
7.6.4 FIFOs operation 128
7.6.5 Data Flow Control 129
7.6.6 PCM Serial Clocks and Sampling Frequencies 130
7.6.7 Interrupts 130
8 Internal CODEC 131
8.1 Overview 131
8.2 Features 131
8.2.1 Signal Descriptions 132
8.2.2 Block Diagram 133
8.2.3 Application schematic 134
8.3 Mapped Register Descriptions 134
8.3.1 CODEC internal register access control (RGADW) 135
8.3.2 CODEC internal register data output (RGDATA) 136

8.4 Operation 136
8.4.1 Access to internal registers of the embedded CODEC 137
8.4.2 CODEC controlling and typical operations 137
8.4.3 Power saving 138
8.4.4 Pop noise and the reduction of it 138
8.5 Timing parameters 139
8.6 C parameters 140
8.7 CODEC internal Registers 140
8.7.1 Registers Memory Map 141
8.7.2 Register Description 143
8.8 Programmable gains 174
8.8.1 Programmable boost gain: GIM 174
8.8.2 Programmable input gain amplifier: GID 175
8.8.3 Programmable digital attenuation: GOD 175
8.8.4 Programmable attenuation: GO 176
8.8.5 Programmable digital mixer gain: GIMIX and GOMIX 176
8.8.6 Gain refresh strategy 177
8.9 Configuration of the headphone output stage 177
8.10 Out-of-band noise filtering 178
8.10.1 Reset of short circuit detection 178
8.11 Sampling frequency: FREQ 178
8.12 Programmable data word length 179
8.13 Ramping system note 179
8.14 AGC system guide 180
8.14.1 AGC operating mode 180
8.15 DRC description 182
8.16 Digital Mixer description 183
8.17 Digital microphone interface 184
8.17.1 Timing Diagram 185
8.17.2 Timings 185
8.17.3 Noise template (TBC) 186
8.17.4 Power Supply Noise Tolerance Template (PSNT2) on analog power supply for I/O buffers of DAC outputs (VDDIO_CODEC) 186
8.18 CODEC Operating modes 187
8.18.1 Initial all the gain 188
8.18.2 Soft Mute mode 189
8.18.3 Power-down and sleep modes 190
8.18.4 Working modes summary 191
8.19 MCLK turn-off and turn-on 192
8.20 Requirements on outputs and inputs selection and power-down modes 192
8.20.1 Initialization and configuration 192
8.21 Circuits design suggestions 193
8.21.1 Avoid quiet ground common currents 193

8.21.2 Microphone connection 194
8.21.3 PCB considerations 196
8.22 Analog characteristics 198
8.22.1 Operating conditions 198
8.22.2 With PWM output, used for analog amplifier application 198
8.22.3 Microphone / Line input to ADC path 199
8.22.4 Micbias and reference 201
8.22.5 I/O buffers 201
8.22.6 Characteristics of the ADC High Pass Filter 202
8.22.7 Characteristics of the ADC Wind Noise Filter 202

Section 5 Memory Interface

9 DDR Controller 204
9.1 Overview 204
9.1.1 Supported DDR SDRAM Types 204
9.1.2 Block Diagram 204
9.2 Register Description 205
9.2.1 DSTATUS 208
9.2.2 DCFG 210
9.2.3 DCTRL 212
9.2.4 DLMR 215
9.2.5 DTIMING1,2,3,4,5,6 (DDR Timing Configure Register) 216
9.2.6 DREFCNT (DDR Auto-Refresh Counter) 220
9.2.7 DMMAP0,1 (DDR Memory Map Configure Register) 221
9.2.8 DDLP (DDR DFI low power handshake control register) 222
9.2.9 DREMAP1,2,3,4,5 (DDR Address Remapping Register 1,2,3,4,5) 222
9.2.10 WCMDCTRL1 (Performance wcmd reorder & grouping) 224
9.2.11 RCMDCTRL0 (Performance rcmd request control) 225
9.2.12 RCMDCTRL1 (Performance rcmd request control) 226
9.2.13 BOUNDARYSEL (Channel boundary select) 227
9.2.14 WDATTHD0 (performance wcmd request control) 229
9.2.15 WDATTHD1 (performance wcmd request control) 229
9.2.16 IPORTPRI (performance priority control) 230
9.2.17 CHxQOS0,1,2,3,4,5 (performance QoS control) 231
9.2.18 AUTOSR_CNT 231
9.2.19 AUTOSR_EN 232
9.2.20 CLKSTP_CFG 232
9.2.21 DDRC_STATUS 233
9.2.22 PHYRET_CFG 233
9.2.23 PHYRST_CFG 234
9.2.24 CPM_DRCG 234
9.3 Functional Description 235

9.3.1 DDRC and DDR2 Memory Initialization Sequence 235
9.4 Change Clock Frequency 236
9.4.1 Manually SELF-REFRESH Mode 236
9.4.2 CPM driven SELF-REFRESH Mode 236
9.4.3 DLL bypass mode 236
9.5 Data Endian 237
9.6 Retention Flow 237
9.6.1 Enter Retention mode 237
9.6.2 Exit Retention mode 238

10 SPI Flash Controller(SFC) 242

10.1 Overview 242
10.2 Features 242
10.3 Block Diagram 243
10.4 Functional Description 243
10.4.1 Bus function 243
10.4.2 Configurable Timing parameter 244
10.5 Pin Description 245
10.6 Data Format Description 245
10.6.1 Transfer format 245
10.6.2 Endian Description 246
10.7 Registers Description 246
10.7.1 Instructions 246
10.7.2 Map 247
10.7.3 Registers 248
10.8 Software Guideline 258
10.8.1 Phase Description 258
10.8.2 Multi phases flow 259
10.8.3 One phase flow 259
10.8.4 Meters of DMA operation need attention 260
10.8.5 Meters of slave mode operation need attention 260
10.8.6 Example (NAND flash write with multi phases in DMA mode) 260
10.8.7 Example (NAND flash write with single phase in REG mode) 261
10.9 Index 263

Section 6 System Functions

11 Clock Reset and Power Controller 265

11.1 Overview 265
11.1.1 CGU Block Diagram 266
11.1.2 CGU Registers 267
11.1.3 PLL Operation 292
11.1.4 Main Clock Division Change Sequence 294

11.2 Power Manager 294
11.2.1 Low-Power Modes and Function 294
11.2.2 Register Description 295
11.2.3 IDLE Mode 303
11.2.4 SLEEP Mode 304
11.3 Reset Control Module 304
11.3.1 Register Description 304
11.3.2 Power On Reset 305
11.3.3 WDT Reset 305
12 Timer/Counter Unit 306
12.1 Overview 306
12.2 Pin Description 306
12.3 Register Description 307
12.3.1 Timer Control Register (TCSR) 308
12.3.2 Timer Data FULL Register (TDFR) 311
12.3.3 Timer Data HALF Register (TDHR) 311
12.3.4 Timer Counter (TCNT) 311
12.3.5 Timer Counter Enable Register (TER) 312
12.3.6 Timer Counter Enable Set Register (TESR) 313
12.3.7 Timer Counter Enable Clear Register (TECR) 314
12.3.8 Timer Flag Register (TFR) 315
12.3.9 Timer Flag Set Register (TFSR) 315
12.3.10 Timer Flag Clear Register (TFCR) 316
12.3.11 Timer Mast Register (TMR) 317
12.3.12 Timer Mask Set Register (TMSR) 318
12.3.13 Timer Mask Clear Register (TMCR) 318
12.3.14 Timer Stop Register (TSR) 319
12.3.15 Timer Stop Set Register (TSSR) 320
12.3.16 Timer Stop Clear Register (TSCR) 321
12.3.17 Timer Status Register (TSTR) 322
12.3.18 Timer Status Set Register (TSTSR) 323
12.3.19 Timer Status Clear Register (TSTCR) 323
12.4 Operation 324
12.4.1 Basic Operation in TCU1 Mode 324
12.4.2 Disable and Shutdown Operation in TCU1 Mode 325
12.4.3 Basic Operation in TCU2 Mode 325
12.4.4 Disable and Shutdown Operation in TCU2 Mode 325
12.4.5 Read Counter in TCU2 Mode 325
12.4.6 Pulse Width Modulator (PWM) 326
12.4.7 Trackball Input Waveform Detect 326
13 Operating System Timer 328

13.1 overview 328
13.2 register description 328
13.2.1 Timer Clock Control Register (OSTCCR) 329
13.2.2 Timer Counter Enable Register (OSTER, OSTESR, OSTECR) 330
13.2.3 Timer Counter Clear Register (OSTCR) 331
13.2.4 Timer Data FULL Register (OST1DFR) 332
13.2.5 Timer Counter (OST1CNT) 332
13.2.6 Timer Flag Register (OST1FR) 332
13.2.7 Timer Mask Register (OST1MR) 333
13.2.8 Operating System Timer Counter (OST2CNTH, OST2CNTL) 333
13.2.9 Operating System Timer Counter high 32 bits buffer (OSTCNT2HBUF) 333
13.3 Operation 334
13.3.1 Basic Operation in channel1 334
13.3.2 Stop Operation in channel1 334
13.3.3 modify the prsecale in channel1 334

14 Interrupt Controller 335

14.1 Overview 335
14.2 Register Description 335
14.2.1 Interrupt Controller Source Register (ICSR0) 336
14.2.2 Interrupt Controller Source Register (ICSR1) 337
14.2.3 Interrupt Controller Mask Register (ICMR0) 337
14.2.4 Interrupt Controller Mask Register (ICMR1) 338
14.2.5 Interrupt Controller Mask Set Register (ICMSR0) 338
14.2.6 Interrupt Controller Mask Set Register (ICMSR1) 338
14.2.7 Interrupt Controller Mask Clear Register (ICMCR0) 339
14.2.8 Interrupt Controller Mask Clear Register (ICMCR1) 339
14.2.9 Interrupt Controller Pending Register (ICPR0) 339
14.2.10 Interrupt Controller Pending Register (ICPR1) 340
14.2.11 Interrupt Source Register0 for PDMA (DSR0) 340
14.2.12 Interrupt Mask Register0 for PDMA (DMR0) 341
14.2.13 Interrupt Pending Register0 for PDMA (DPR0) 341
14.2.14 Interrupt Source Register1 to PDMA (DSR1) 341
14.2.15 Interrupt Mask Register1 for PDMA (DMR1) 342
14.2.16 Interrupt Pending Register1 for PDMA (DPR1) 342
14.3 Software Considerations 343

15 Watchdog Timer 344

15.1 Overview 344
15.2 Register Description 344
15.2.1 Watchdog Control Register (TCSR) 345
15.2.2 Watchdog Enable Register (TCER) 346
15.2.3 Watchdog Timer Data Register (TDR) 346

15.2.4 Watchdog Timer Counter (TCNT) 347
15.3 Watchdog Timer Function 347
16 PDMA Controller 348
16.1 Overview 348
16.2 Features 348
16.3 Block Diagram 348
16.4 Memory Mapped Register Descriptions 349
16.4.1 DMA Channel Registers 349
16.4.2 Global Control Registers 349
16.5 DMA Channel Register Definition 350
16.5.1 DMA Source Address (DSAn, n = 0 ~ 7) 350
16.5.2 DMA Target Address (DTAn, n = 0 ~ 7) 351
16.5.3 DMA Transfer Count (DTCn, n = 0 ~ 7) 351
16.5.4 DMA Request Types (DRTn, n = 0 ~ 7) 351
16.5.5 DMA Channel Control/Status (DCSn, n = 0 ~ 7) 352
16.5.6 DMA Channel Command (DCMn, n = 0 ~ 7) 353
16.5.7 DMA Descriptor Address (DDAn, n = 0 ~ 7) 356
16.5.8 DMA Stride Difference (DSDn, n = 0 ~ 7) 356
16.6 DMA Global Register Definition 357
16.6.1 DMA Control 357
16.6.2 DMA Interrupt Pending (DIRQP) 358
16.6.3 DMA Doorbell (DDB) 358
16.6.4 DMA Doorbell Set (DDS) 359
16.6.5 Descriptor Interrupt Pending (DIP) 359
16.6.6 Descriptor Interrupt Clear (DIC) 359
16.6.7 DMA Channel Programmable (DMACP) 360
16.6.8 DMA Soft IRQ Pending (DSIRQP) 360
16.6.9 DMA Soft IRQ Mask (DSIRQM) 361
16.6.10 DMA Channel IRQ Pending to MCU (DCIRQP) 361
16.6.11 DMA Channel IRQ to MCU Mask (DCIRQM) 362
16.7 MCU 362
16.7.1 MCU Control & Status 362
16.7.2 MCU Normal MailBox 363
16.7.3 MCU Security MailBox 363
16.7.4 MCU Interrupt 364
16.7.5 Multiple Bank Tightly Coupled Sharing Memory 364
16.7.6 CP0 Registers of MCU 364
16.7.7 Normal Exceptions Accepted by MCU 365
16.7.8 How to Boot MCU Up 365
16.8 DMA manipulation 366
16.8.1 Descriptor Transfer Mode 366
16.8.2 No-Descriptor Transfer Mode 369

16.8.3 Descriptor Transfer Interrupt/Stop control 369
16.9 DMA Requests 371
16.9.1 Auto Request 371
16.9.2 On-Chip Peripheral Request 371
16.10 How to Use Programmable DMA Channel 371
17 Real Time Clock 372
17.1 Overview 372
17.2 Features 372
17.3 Block Diagram 372
17.4 Pins Description 373
17.5 Registers Description 373
17.5.1 RTC Control Register (RTCCR) 375
17.5.2 RTC Second Register (RTCSR) 376
17.5.3 RTC Second Alarm Register (RTCSAR) 377
17.5.4 RTC Regulator Register (RTCGR) 377
17.5.5 Hibernate Control Register (HCR) 378
17.5.6 HIBERNATE mode Wakeup Filter Counter Register (HWFCR) 378
17.5.7 Hibernate Reset Counter Register (HRCR) 379
17.5.8 HIBERNATE Wakeup Control Register (HWCR) 379
17.5.9 HIBERNATE Wakeup Status Register (HWRSR) 380
17.5.10 Hibernate Scratch Pattern Register (HSPR) 381
17.5.11 Write Enable Pattern Register (WENR) 382
17.5.12 WKUP_PIN_RST control register (WKUPPINCR) 382
17.6 Operation Flow 383
17.6.1 Registers Access 383
17.6.2 Normal Mode 384
17.6.3 HIBERNATE Mode 384
17.6.4 Time Regulation 385
17.6.5 Clock select 385
18 EFUSE Slave Interface (EFUSE) 387
18.1 Overview 387
18.2 Registers 387
18.2.1 Registers Memory Map 388
18.2.2 Registers and Fields Description 388
18.3 Flow 392
18.3.1 Program EFUSE Flow 392
18.3.2 Program Security Key Flow 393
18.3.3 Read EFUSE Flow 393
18.3.4 Read Security Key/Random Number Flow 393

Section 7 Peripherals

19 General-Purpose I/O Ports 395
19.1 Overview 395
19.2 Features 395
19.3 About GPIO Port Summary Table 395
19.3.1 GPIO Port A Summary 396
19.3.2 GPIO Port B Summary 397
19.3.3 GPIO Port C Summary 398
19.3.4 GPIO Port D Summary 398
19.3.5 GPIO Port Z - Shadow Group 398
19.4 Registers Description 399
19.4.1 PORT A Register Group 402
19.4.2 PORT B Register Group 410
19.4.3 PORT C Register Group 417
19.4.4 PORT D Register Group 430
19.4.5 PORT Z Shadow Register Group 438
19.4.6 GPIOZ Group ID to Load Register (PzGID2LD,0xF0) 440
19.5 Program Guide 441
19.5.1 Port Function Guide 441
19.5.2 Configure without 3rd-unexpected state 441
20 SMB Controller 443
20.1 Overview 443
20.1.1 Features 443
20.1.2 Pin Description 443
20.2 Registers 444
20.2.1 Registers Memory Map 444
20.2.2 Registers and Fields Description 445
20.3 Operating Flow 469
20.3.1 SMB Behavior 470
20.3.2 Master Mode Operation 470
20.3.3 Slave Mode Operation 472
20.3.4 Disabling SMB 475
20.3.5 Summary the condition could flush TX FIFO 476
20.3.6 The condition could generate START, STOP and RESTART 476
21 Smart Card Controller 480
21.1 Overview 480
21.2 Pin Description 481
21.3 Register Description 481
21.3.1 Transmit/Receive FIFO Data Register (SCCDR) 482
21.3.2 FIFO Data Count Register (SCCFDR) 482
21.3.3 Control Register (SCCCR) 482

21.3.4 Status Register (SCCSR) 484
21.3.5 Transmission Factor Register (SCCTFR) 484
21.3.6 Extra Guard Timer Register (SCCEGTR) 485
21.3.7 ETU Counter Value Register (SCCECR) 485
21.3.8 Reception Timeout Register (SCCRTOR) 485
22 Synchronous Serial Interface(SSI) 486
22.1 Overview 486
22.2 Features: 486
22.3 Block Diagram 487
22.4 Functional Description 487
22.5 Pin Description 488
22.6 Data Formats 488
22.6.1 Motorola's SPI Format Details 489
22.6.2 TI's SSP Format Details 492
22.6.3 National Microwire Format Details 493
22.7 Register Description 495
22.7.1 Register Mapping 495
22.7.2 SSI Data Register (SSIDR) 496
22.7.3 SSI Control Register0 (SSICR0) 497
22.7.4 SSI Control Register1 (SSICR1) 499
22.7.5 SSI Status Register (SSISR) 501
22.7.6 SSI Interval Time Control Register (SSIITR) 502
22.7.7 SSI Interval Character-per-frame Control Register (SSIICR) 503
22.7.8 SSI Clock Generator Register (SSIGR) 503
22.7.9 SSI Receive Counter Register (SSIRCNT) 504
22.8 Software Guideline 504
22.8.1 Common flow 504
22.8.2 Interrupt Operation 505
22.9 Index 505
23 Universal Asynchronous Receiver/Transmitter(uart) 507
23.1 Overview 507
23.2 Features 507
23.3 Block Diagram 507
23.4 Functional Description 508
23.4.1 Full-duplex operation 508
23.4.2 The meaning of all bits 508
23.4.3 RFIFO and TFIFO 508
23.4.4 Transmission, reception and line status Independently 508
23.4.5 Slow infrared asynchronous interface 508
23.5 Pins Description 508
23.6 Data Format Description 509

23.7 Register Description 509
23.7.1 Register Memory Map 509
23.7.2 Register and Fields Description 510
23.8 Operation Flow 524
23.8.1 UART Configuration 524
23.8.2 Data Transmission 524
23.8.3 Data Reception 524
23.8.4 Receive Error Handling 525
23.8.5 Modem Transfer 525
23.8.6 DMA Transfer 525
23.8.7 Slow IrDA Asynchronous Interface 525
23.8.8 For any frequency clock to use the UART 526
23.9 Index 528
24 MMC/SD CE-ATA Controller（MSC） 529
24.1 Overview 529
24.2 Features 529
24.3 Block Diagram 530
24.4 Functional Description 530
24.4.1 MSC Reset 531
24.4.2 Voltage Validation 531
24.4.3 Card Registry 531
24.4.4 Card Access 532
24.4.5 Protection Management 534
24.4.6 Card Status 537
24.4.7 SD Status 541
24.4.8 SDIO 541
24.4.9 Clock Control 543
24.4.10 Application Specified Command Handling 543
24.5 Pins Description 543
24.6 Data Format Description 544
24.7 Register Description 544
24.7.1 Register Memory Map 545
24.7.2 Register and Fields Description 546
24.8 Operation Flow 566
24.8.1 Data FIFOs 566
24.8.2 DMA and Program I/O 566
24.8.3 Start and Stop clock 568
24.8.4 Software Reset 569
24.8.5 Voltage Validation and Card Registry 569
24.8.6 Single Data Block Write 571
24.8.7 Single Block Read 572
24.8.8 Multiple Block Write 572

24.8.9 Multiple Block Read 573
24.8.10 Stream Write (MMC) 574
24.8.11 Stream Read (MMC) 575
24.8.12 Erase, Select/Deselect and Stop 575
24.8.13 SDIO Suspend/Resume 576
24.8.14 SDIO Read Wait 576
24.8.15 Operation and Interrupt 576
24.9 Index 577

25 OTG Controller 579

25.1 Overview 579
25.2 Block Diagram 580
25.2.1 Slave-Only mode 580
25.2.2 Internal DMA mode 580
25.3 Pin Description 580
25.4 Register Map 581
25.4.1 CSR Memory Map 581
25.4.2 Register Maps 581
25.4.3 Global CSR Map 582
25.4.4 Host Mode CSR Map 583
25.4.5 Device Mode CSR Map 583
25.4.6 Data FIFO (DFIFO) Access Register Map 585
25.5 Register Descriptions 586
25.5.1 Application Access to the CSRs 586
25.5.2 Overview of Commonly Used Register Bits 587
25.5.3 Global Registers 591
25.5.4 Host Mode Registers 629
25.5.5 Device Mode Registers 647
25.6 Operation Flow 680
25.6.1 Core Initialization 680
25.6.2 Programming the Device Core 681
25.6.3 Programming the Host Core 684

26 MAC 686

26.1 Features 686

Section 8 Boot

27 XBurst Boot ROM Specification 688

27.1 Boot Select 688
27.2 Boot Procedure 688
27.3 SPL Structure 690
27.4 SPL Paramaters 690

27.5 USB Boot Specification 694
27.6 MSC0 Boot Specification 698
27.7 MSC1 boot Specification 700
27.8 SFC boot Specification 700

TABLES

Table 4-1 SLCD Pins Description 31
Table 4-2 Registers Memory Map-Address Base 33
Table 4-3 SLCD Controller Registers Description 33
Table 5-1 Camera Interface Pins Description 62
Table 5-2 CIM Registers 62
Table 5-3 The modes and the corresponding signals used 74
Table 6-1 AIC Pins Description 82
Table 6-2 AIC Registers Description 83
Table 6-3 Sample data bit relate to SDATA_IN/SDATA_OUT bit 101
Table 6-4 Audio Sampling rate, BIT_CLK and SYS_CLK frequencies 111
Table 6-5 BIT_CLK divider setting 112
Table 6-6 Approximate common multiple of SYS_CLK for all sample rates 113
Table 6-7 CPM/AIC clock divider setting for various sampling rate if PLL = 270.64MHz 113
Table 6-8 PLL parameters and audio sample errors for EXCLK=12MHz 114
Table 7-1 PCM Interface Pins Description 117
Table 7-2 Registers Memory Map-Address Base 119
Table 7-3 PCM Registers Description 119
Table 8-1 CODEC signal IO pin description 132
Table 8-2 Internal CODEC Mapped Registers Description (AIC Registers) 135
Table 8-3 Microphone/Line input performances 200
Table 8-4 I/O buffer static characteristics 201
Table 8-5 I/O buffer dynamic characteristics 201
Table 9-1 DDRC Register 205
Table 10-1 I/O Pin Description 245
Table 10-2 SFC Registers Map 247
Table 10-3 Terms and Abbreviations 263
Table 11-1 Registers Memory Map-Address Base 267
Table 11-2 CGU Registers Configuration 267
Table 11-3 Power/Reset Management Controller Registers Configuration 295
Table 12-1 PWM Pins Description 306
Table 12-2 Registers Memory Map-Address Base 307
Table 12-3 TCU Registers Configuration 307
Table 13-1 Registers Memory Map-Address Base 328
Table 13-2 SYS_OST Registers Configuration 329
Table 14-1 Registers Memory Map-Address Base 336
Table 14-2 INTC Register 336
Table 15-1 Registers Memory Map-Address Base 344
Table 15-2 WDT Registers Configuration 345
Table 16-1 Registers Memory Map-Address Base 349
Table 16-2 DMA Channel Registers (n=0~31) 349
Table 16-3 Registers Memory Map-Address Base 350

Table 16-4 Global Control Registers 350
Table 16-5 Transfer Request Types 352
Table 16-6 Available RDIL 354
Table 16-7 TCSM space 364
Table 16-8 Descriptor Structure 367
Table 17-1 Registers Memory Map-Address Base 374
Table 17-2 Registers for real time clock 374
Table 17-3 Registers for hibernating mode 374
Table 17-4 Clock select registers 386
Table 19-1 GPIO port summary table illustration 395
Table 19-2 GPIO Port A summary 396
Table 19-3 GPIO Port B summary 397
Table 19-4 GPIO Port C summary 398
Table 19-5 GPIO Port D summary 398
Table 19-6 Registers Memory Map-Address Base 399
Table 19-7 GPIO Registers 400
Table 20-1 SMB Pin Description 443
Table 20-2 Registers Memory Map-Address Base 444
Table 20-3 Registers Memory Map-Address Offset 444
Table 21-1 Smart Card Controller Pins Description 481
Table 21-2 Smart Card Controller Registers Description 481
Table 22-1 SSI Controller Pins Description 488
Table 22-2 SSI Serial Port Registers 496
Table 22-3 SSI Interrupts 505
Table 23-1 UART Pins Description 508
Table 23-2 Registers Memory Map Base Address 509
Table 23-3 UART Registers Map 510
Table 23-4 UART Interrupt Identification Register Description 514
Table 23-5 Description of Proprietary Vocabulary 528
Table 24-1 Command Data Block Structure 535
Table 24-2 Card Status Description 538
Table 24-3 SD Status Structure 541
Table 24-4 Registers Memory Map Base Address 545
Table 24-5 MSC Registers Map 545
Table 24-6 How to stop multiple block write 573
Table 24-7 How to stop multiple block read 573
Table 24-8 Description of Proprietary Vocabulary 577
Table 24-9 The mapping between Commands and Steps 577
Table 25-1 OTG Pins Description 580
Table 25-2 Registers Memory Map Base Address 581
Table 25-3 Core Global CSR Map (000h-3FFh) 582
Table 25-4 Host Mode CSR Map (400h-7FFh) 583
Table 25-5 Device Mode CSR Map (800h-BFFh) 583

Table 25-6 Data FIFO (DFIFO) Access Register Map 585
Table 25-7 List of Commonly Used Register Bits 587
Table 25-8 Control and Status Register: GOTGCTL 591
Table 25-9 Interrupt Register: GOTGINT 595
Table 25-10 AHB Configuration Register: GAHBCFG 596
Table 25-11 USB Configuration Register: GUSBCFG 600
Table 25-12 Reset Register: GRSTCTL 605
Table 25-13 Interrupt Register: GINTSTS 608
Table 25-14 Interrupt Mask Register: GINTMSK 615
Table 25-15 Host Mode Receive Status Debug Read/Status Read and Pop Registers: GRXSTSR/GRXSTSP 617
Table 25-16 Device Mode Receive Status Debug Read/Status Read and Pop Registers: GRXSTSR/GRXSTSP 617
Table 25-17 Receive FIFO Size Register: GRXFSIZ 618
Table 25-18 Non-Periodic Transmit FIFO Size Register: GNPTXFSIZ (Host Mode and Device Shared FIFO Mode) 619
Table 25-19 Non-Periodic Transmit FIFO Size Register: GNPTXFSIZ (Device Dedicated FIFO Mode) 619
Table 25-20 Non-Periodic Transmit FIFO/Queue Status Register: GNPTXSTS 619
Table 25-21 User HW Config1 Register: GHWCFG1 620
Table 25-22 User HW Config2 Register: GHWCFG2 621
Table 25-23 User HW Config3 Register: GHWCFG3 623
Table 25-24 User HW Config4 Register: GHWCFG4 625
Table 25-25 Global DFIFO Software Config Register: GDFIFOCFG 627
Table 25-26 Host Periodic Transmit FIFO Size Register: HPTXFSIZ 628
Table 25-27 Device Periodic Transmit FIFO-n Register: DPTXFSIZn 628
Table 25-28 Device In Endpoint Transmit FIFO Size Register: (DIEPTXFn) 629
Table 25-29 Host Configuration Register: HCFG 630
Table 25-30 Host Frame Interval Register: HFIR 632
Table 25-31 Host Frame Number/Frame Time Remaining Register: HFNUM 633
Table 25-32 Host Periodic Transmit FIFO/Queue Status Register: HPTXSTS 633
Table 25-33 Host All Channels Interrupt Register: HAINT 634
Table 25-34 Host All Channels Interrupt Mask Register: HAINTMSK 634
Table 25-35 Host Port Control and Status Register: HPRT 635
Table 25-36 Host Channel-n Characteristics Register: HCCHARn 638
Table 25-37 Host Channel-n Split Control Register: HCSPLTn 640
Table 25-38 Host Channel-n Interrupt Register: HCINTn 641
Table 25-39 Host Channel-n Interrupt Mask Register: HCINTMSKn 643
Table 25-40 Host Channel-n Transfer Size Register: HCTSIZn 644
Table 25-41 Host Channel-n DMA Address Register: HCDMAn 646
Table 25-42 Host Channel-n DMA Buffer Address Register: HCDMABn 647
Table 25-43 Host Frame List Base Address Register: HFLBAddr 647
Table 25-44 Device Configuration Register: DCFG 648

Table 25-45 device Control Register: DCTL 650
Table 25-46 Minimum Duration for Soft Disconnect 654
Table 25-47 Device Status Register: DSTS 654
Table 25-48 Device IN Endpoint Common Interrupt Mask Register: DIEPMSK 656
Table 25-49 Device OUT Endpoint Common Interrupt Mask Register: DOEPMSK 656
Table 25-50 Device All Endpoints Interrupt Register: DAINT 657
Table 25-51 Device Endpoints Interrupt Mask Register: DAINTMSK 658
Table 25-52 Device IN Token Sequence Learning Queue Read Register 1: DTKNQR1 658
Table 25-53 Device IN Token Sequence Learning Queue Register 2: DTKNQR2 659
Table 25-54 Device IN Token Sequence Learning Queue Register 3: DTKNQR3 659
Table 25-55 Device IN Token Sequence Learning Queue Register 4: DTKNQR4 659
Table 25-56 Device VBUS Discharge Time Register: DVBUSDIS 660
Table 25-57 Device VBUS Pulsing Time Register (DVBUSPULSE) 660
Table 25-58 Device Threshold Control Register (DTHRCTL) 661
Table 25-59 Device IN Endpoint FIFO Empty Interrupt Mask Register: DIEPEMPMSK 662
Table 25-60 Device Each Endpoint Interrupt Register: DEACHINT 663
Table 25-61 Device Each Endpoint Interrupt Register Mask: DEACHINTMSK 663
Table 25-62 Device Each In Endpoint-n Interrupt Register: DIEPEACHMSKn 664
Table 25-63 Device Each Out Endpoint-n Interrupt Register: DOEPEACHMSKn 665
Table 25-64 Device Control IN Endpoint 0 Control Register: DIEPCTL0 665
Table 25-65 Device OUT Endpoint 0 Control Register: DOEPCTL0 667
Table 25-66 Device Endpoint-n Control Register: DIEPCTLn/DOEPCTLn 669
Table 25-67 Device Endpoint-n Interrupt Register: DIEPINTn/DOEPINTn 673
Table 25-68 Device IN Endpoint 0 Transfer Size Register: DIEPTSIZ0 676
Table 25-69 Device OUT Endpoint 0 Transfer Size Register: DOEPTSIZ0 677
Table 25-70 Device Endpoint-n Transfer Size Register: DIEPTSIZn/DOEPTSIZn 677
Table 25-71 Device Endpoint-n DMA Address Register: DIEPDMAn/DOEPDMAn 679
Table 25-72 Device Endpoint-n DMA Buffer Address Register: DIEPDMABn/DOEPDMABn 679
Table 25-73 Device IN Endpoint Transmit FIFO Status Register: DTXFSTSn 680
Table 27-1 Boot Configuration of X1000 688
Table 27-2 SPL parameters structure 691
Table 27-3 Transfer Types Used by the Boot Program 694
Table 27-4 Vendor Request 0 Setup Command Data Structure 697
Table 27-5 Vendor Request 1 Setup Command Data Structure 697
Table 27-6 Vendor Request 2 Setup Command Data Structure 697
Table 27-7 Vendor Request 3 Setup Command Data Structure 697
Table 27-8 Vendor Request 4 Setup Command Data Structure 698
Table 27-9 Vendor Request 5 Setup Command Data Structure 698
Table 27-10 SPI NOR flash boot flag informations(in spl signature) 702
Table 27-11 SPI NAND flash boot flag informations(in spl signature) 702

FIGURES

Figure 1-1 X1000 Diagram 2
Figure 2-1 Structure of CPU core 12
Figure 4-1 Block Diagram of SLCD 32
Figure 5-1 ITU656 Progressive Mode 76
Figure 6-1 AIC Block Diagram 80
Figure 6-2 Interface to an External Master Mode I2S/MSB-Justified CODEC Diagram 81
Figure 6-3 Interface to an External Slave Mode I2S/MSB-Justified CODEC Diagram 81
Figure 6-4 Interface to a HDMI Transmitter via SPDIF Diagram 82
Figure 6-5 I2S data format (A: LR mode) 99
Figure 6-6 I2S data format (B: RL mode) 100
Figure 6-7 MSB-justified data format (C: LR mode) 100
Figure 6-8 MSB-justified data format (D: RL mode) 100
Figure 6-9 Block format 102
Figure 6-10 Sub-frame format in PCM mode 102
Figure 6-11 Sub-frame format in non-PCM mode 103
Figure 6-12 Transmitting/Receiving FIFO access via APB Bus 106
Figure 6-13 One channel (Left) and Two channels (right) mode (16 bits packed mode) 108
Figure 6-14 Four channels (Left) and Six channels (right) mode (16 bits packed mode) 109
Figure 6-15 Eight channels mode (16 bits packed mode) 109
Figure 6-16 One channel (Left) and Two channels (right) mode 109
Figure 6-17 Four channels (Left) and Six channels (right) mode 110
Figure 6-18 Eight channels mode 110
Figure 6-19 SYS_CLK, BIT_CLK and SYNC generation scheme 112
Figure 7-1 Short Frame SYN Timing (Shown with 16bit Sample) 125
Figure 7-2 Short Frame SYN Timing (Shown with 16bit Sample) 125
Figure 7-3 Long Frame SYN Timing (Shown with 16bit Sample) 126
Figure 7-4 Long Frame SYN Timing (Shown with 16bit Sample) 126
Figure 7-5 Multi-Slot Frame SYN Timing (Shown with two Slots and 8bit Sample) 126
Figure 7-6 Transmitting/Receiving FIFO access via APB Bus 129
Figure 7-7 PCMCLK and PCMSYN generation scheme 130
Figure 8-1 CODEC block diagram 133
Figure 8-2 Internal CODEC works with AIC 133
Figure 8-3 AGC adjusting waves 181
Figure 8-4 AGC adjust areas 182
Figure 8-5 signal above threshold at the output of the soft clipping DRC 183
Figure 8-6 Digital Mixer structure 183
Figure 8-7 Digital microphone interface connection 185
Figure 8-8 Digital microphone timing diagram at MCLK = 12 MHz (DMIC_CLK = 3 MHz) 185
Figure 8-9 Digital microphone modulation noise reference spectrum (with FFT resolution = 20 Hz and 7 terms Blackman-Harris windowing) 186
Figure 8-10 PSNT2 for VDDIO_CODEC when using PWM output 187

Figure 8-11 PSNT2 for VDDIO_CODEC when using PWM output 187
Figure 8-12 CODEC Power Diagram 188
Figure 8-13 ADC Gain up and gain down sequence 189
Figure 8-14 DAC Gain up and gain down sequence 190
Figure 8-15 Peripheral power supply connection 193
Figure 8-16 Differential routing 196
Figure 8-17 PCB optimization for avoiding EMI issue 197
Figure 9-1 DDR system block diagram 205
Figure 10-1 SFC Block Diagram 243
Figure 10-2 Sample point(no delay) 244
Figure 10-3 Sample point(has delay) 244
Figure 10-4 AC timing 244
Figure 10-5 Data Format(Standard SPI) 245
Figure 10-6 Data Format(Dual SPI) 245
Figure 10-7 Data Format(Quad SPI) 246
Figure 10-8 Multi phases flow 259
Figure 10-9 One phase flow 260
Figure 11-1 Block Diagram of PLL 293
Figure 16-1 Block Diagram of PDMA 348
Figure 16-2 Descriptor Transfer Flow 368
Figure 16-3 Example for Stride Transfer Mode 369
Figure 17-1 Core Power On 384
Figure 17-2 RTC clock selection path 386
Figure 20-1 Master Transmitter — Tx FIFO Empties/STOP Generation 476
Figure 20-2 Master Receiver — Tx FIFO Empties/STOP Generation 477
Figure 20-3 Master Transmitter — Restart bit of SMB_DC is set 477
Figure 20-4 Master Receiver — Restart bit of SMB_DC is set 477
Figure 20-5 Master Transmitter — Stop Bit of SMB_DC Set/Tx FIFO Not Empty 478
Figure 20-6 Master Transmitter — First Byte Loaded Into Tx FIFO Allowed to Empty, Restart Bit Set 478
Figure 20-7 Master Receiver — Stop Bit of IC_DATA_CMD Set/Tx FIFO Not Empty 478
Figure 20-8 Master Receiver — First Command Loaded After Tx FIFO Allowed to Empty/Restart Bit Set 479
Figure 22-1 SSI Block Diagram 487
Figure 22-2 SPI Single Character Transfer Format (PHA = 0) 489
Figure 22-3 SPI Single Character Transfer Format (PHA = 1) 489
Figure 22-4 SPI Back-to-Back Transfer Format 490
Figure 22-5 SPI Frame Interval Mode Transfer Format (ITFRM = 0, LFST = 0) 491
Figure 22-6 SPI Frame Interval Mode Transfer Format (ITFRM = 1, LFST = 1) 492
Figure 22-7 TI's SSP Single Transfer Format 492
Figure 22-8 TI's SSP Back-to-back Transfer Format 493
Figure 22-9 National Microwire Format 1 Single Transfer 494
Figure 22-10 National Microwire Format 1 Back-to-back Transfer 494

Figure 22-11 National Microwire Format 2 Read Timing 494
Figure 22-12 National Microwire Format 2 Write Timing 495
Figure 23-1 UART DATA Controller Block Diagram 507
Figure 23-2 UART Frame Structure Diagram 509
Figure 24-1 MMC/SD Controller Block Diagram 530
Figure 24-2 msc data format 544
Figure 25-1 OTG Slave-Only mode 580
Figure 25-2 OTG Internal DMA mode 580
Figure 25-3 OTG CSR Memory Map 581
Figure 27-1 Boot sequence diagram of X1000 689
Figure 27-2 The basic SPL structure 690
Figure 27-3 SPL parameters location 691
Figure 27-4 SPL configuration change Procedure 693
Figure 27-5 USB Communication Flow 694
Figure 27-6 Typical Procedure of USB Boot 696
Figure 27-7 Typical Procedure of MSC0 Boot 699
Figure 27-8 SPL Structure of MSC0/MSC1 Boot 700
Figure 27-9 X1000 SPI Boot Procedure 701

Section 1

OVERVIEW

1 Overview

X1000 is a low power consumption, high performance and high integrated application processor, the application is focus on IoT devices. And it can match the requirements of many other embedded products.

1.1 Block Diagram

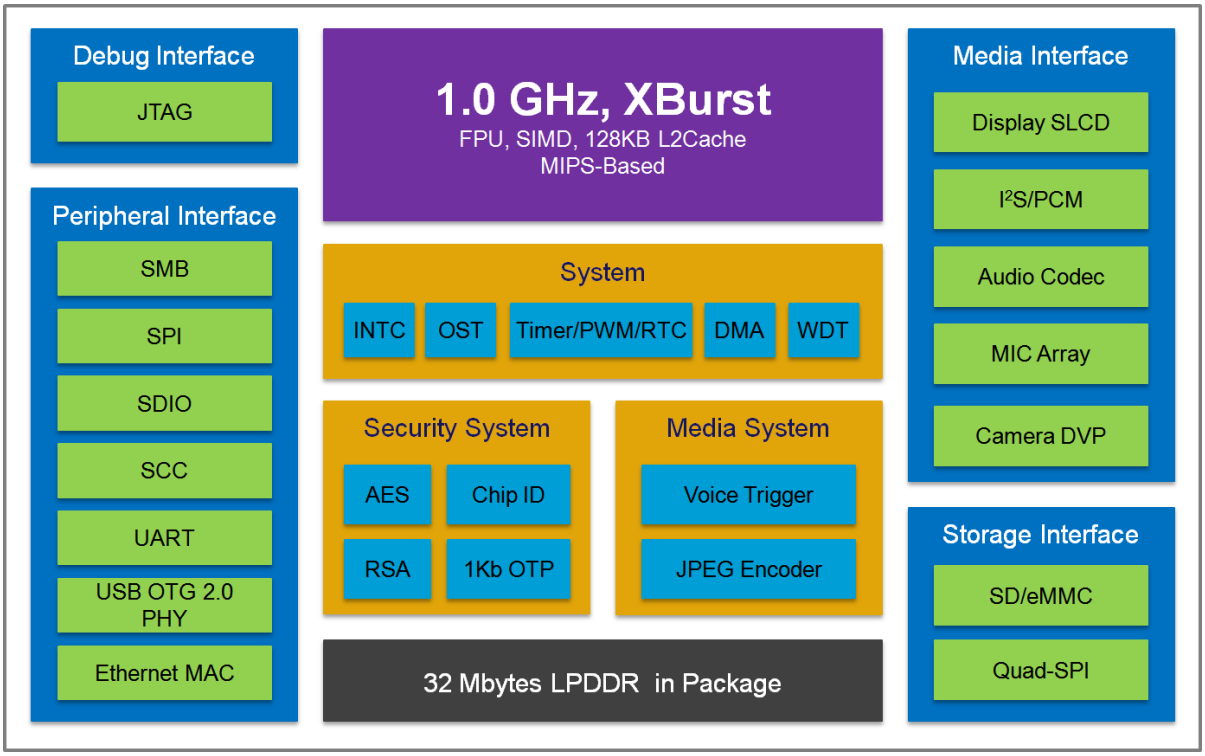


Figure 1-1 X1000 Diagram

1.2 Features

1.2.1 CPU Core

- MIPS-Based XBurst® cores (up to 1.0GHz)
- MIPS-Based XBurst® CPU
  - XBurst® RISC instruction set
  - XBurst® SIMD instruction set
  - XBurst® FPU instruction set supporting both single and double floating point format which are IEEE754 compatible
  - XBurst® 9-stage pipeline micro-architecture
- MMU
  - 32-entry joint-TLB
  - 4 entry Instruction TLB

  - 4 entry data TLB
- L1 Cache
  - 16KB instruction cache
  - 16KB data cache
- Hardware debug support
- 16KB tight coupled memory
- L2 Cache
  - 128KB unify cache
- The XBurst® processor system supports little endian only

1.2.2 Image Core

- Hardware JPEG encoder
  - Baseline ISO/IEC 10918-1 JPEG compliant
  - 8-bit pixel depth support
  - Support for YUY2 ([Y0,U0,Y1,V0]) color
  - Up to four programmable Quantization tables
  - Fully programmable Huffman tables
  - Image size up to 2M pixel

1.2.3 Display/Camera/Audio

- SLCD Controller
  - Basic Features
    - Display size up to 640x480@60Hz,24BPP
  - Colors Supports
    - Support up to 16,777,216 (16M) colors
  - Panel Supports
    - 16bit 8080 once parallel interface
    - 9 bits twice 8080 parallel interface
    - 8 bits twice/third times 8080 parallel
    - Supports different size of display panel
    - Supports internal DMA operation and register operation
- Camera interface module
  - Input image size up to 2M pixels
  - Integrated DMA
  - Supported data format: YCbCr 4:2:2
  - Supports ITU656 (YCbCr 4:2:2) input
  - Configurable VSYNC and HSYNC signals: active high/low
  - Configurable PCLK: active edge rising/falling
  - PCLK max. 30MHz
  - Configurable output order

- AIC controller
  - I2S features

- 8, 16, 18, 20 and 24 bit audio sample data sizes supported, 16 bits packed sample data is supported
- Up to 8 channels sample data supported
- DMA transfer mode supported
- Stop serial clock supported
- Programmable Interrupt function supported
- Support share clock mode and split clock mode.
- Support mono PCM data to stereo PCM data expansion on audio play back
- Support endian switch on 16-bits normal audio samples play back
- Internal programmable or external serial clock and optional system clock supported for I2S or MSB-Justified format
- Internal I2S CODEC supported
- Two FIFOs for transmit and receive respectively

- PCM interface
  - Support master mode and slave mode
  - Data starts with the frame PCMSYN or one PCMCLK later
  - Support three modes of operation for PCM
    - Short frame sync mode
    - Long frame sync mode
    - Multi-slot mode
  - Data is transferred and received with the MSB first
  - The PCM serial output data, PCMDOUT, is clocked out using the rising edge of the PCMSCLK
  - The PCM serial input data, PCMDIN, is clocked in on the falling edge of the PCMSCLK
  - 8/16 bit sample data sizes supported
  - DMA transfer mode supported
  - Two FIFOs for transmit and receive respectively with 16 samples capacity in every direction
- Internal CODEC
  - 24 bits ADC and DAC(digital output)
  - PWM line out and can load down to 16 Ohm
  - Sample rate supported: 8k, 11.025k, 12k, 16k, 22.05k, 24k, 32k, 44.1k, 48k, 88.2k, 96k, 176.4k, and 192k
  - Mono line input
  - DAC(digital output and converter to analog by external circuit): SNR: 95dB A-Weighted, THD: -80dB @FS-1dB
  - Line input to ADC path: SNR: 90dB A-Weighted, THD: -80dB @FS-1dB
  - Separate power-down modes for ADC and DAC path with several shutdown modes
  - Reduction of audible glitches systems: Soft Mute mode
  - Embedded low noise Linear Regulator
  - 1 MIC in path or 1 line in path Maximum (Total 1 analog input)
- Low power DMIC Controller
  - 16 bits data interface and 20bit precision internal controller.

- SNR: 90dB, THD: -90dB @ FS -20dB
- Linear high pass filter include. Attenuation: -2.9dB@100Hz, -22dB@27Hz. -36dB@10Hz
- Low power voice trigger when waiting to start talking.
- 1 to 4 channel MIC support.
- Support voice data pre-fetch when trigger enable and the data interface disable, but do not increase the power dissipation.
- Sample rate supported: 8k, 16k.
- Support low power mode

1.2.4 Memory Interface

- DDR Controller
  - Support LPDDR, DDR2, DDR3
  - 16 bit data width
  - Support size up to 1GB (1 chip select, 3-bit Bank,15-bit Row, 11-bit Column,)
  - Asynchornize to system bus and each port.
  - Support clock-stop mode
  - Support auto self-refresh mode
  - Support power-down mode and deep-power-down mode
  - Programmable DDR timing parameters
  - Programmable DDR row and column address width and order

- 32MB SIP LPDDR

- Serial nand/nor flash interface(SFC)
  - SPI protocol support: Standard, Dual, Quad SPI
  - Standard I/O data transfer up to 80Mbits/s
  - Dual I/O data transfer up to 160Mbits/s
  - Quad I/O data transfer up to 240Mbits/s
  - transmit-only or receive-only operation
  - MSB always be first in intra transfer of one byte. Least Significant Byte first for inter transfer of data bytes, and Most Significant Byte first for inter transfer of command or address bytes.
  - one device select
  - Configurable sampling point for reception
  - Configurable timing parameters: tSLCH, tCHSH and tSHSL
  - Configurable flash address wide are supported
  - 7 transfer formats: Standard SPI, Dual-Output/Dual-Input SPI, Quad-Output/Quad-Input SPI, Dual-I/O SPI, Quad-I/O SPI, Full Dual-I/O SPI, Full Quad-I/O SPI
  - two data transfer mode: slave mode and DMA mode
  - Configurable 6 phases for software flow

1.2.5 System Functions

- Clock generation and power management
  - On-chip oscillator circuit (support 24MHz, 26MHz)
  - Two phase-locked loops (PLL) with programmable multiplier
  - CCLK, HHCLK, H2CLK, PCLK, H0CLK, DDR_CLK frequency can be changed separately for software by setting registers
  - Functional-unit clock gating
  - Supply block power shut down

- Timer and counter unit with PWM output and/or input edge counter
  - Provide 5 channels, all can generate PWM, two of them have input signal transition edge counter
  - 16-bit A counter and 16-bit B counter with auto-reload function every channel
  - Support interrupt generation when the A counter underflows
  - Three clock sources: RTCLK (real time clock), EXCLK (external clock input), PCLK (APB Bus clock) selected with 1, 4, 16, 64, 256 and 1024 clock dividing selected

- OS timer
  - One channel
  - 32-bit counter and 32-bit compare register
  - Support interrupt generation when the counter matches the compare register
  - Three clock sources: RTCLK (real time clock), EXCLK (external clock input), PCLK (APB Bus clock) selected with 1, 4, 16, 64, 256 and 1024 clock dividing selected

- Interrupt controller
  - Total 64 interrupt sources
  - Each interrupt source can be independently enabled
  - Priority mechanism to indicate highest priority interrupt
  - All the registers are accessed by CPU and PDMA
  - Unmasked interrupts can wake up the chip in sleep mode
  - Another set of source, mask and pending registers to serve for PDMA

- Watchdog timer
  - Generates WDT reset
  - A 16-bit Data register and a 16-bit counter
  - Counter clock uses the input clock selected by software
    - PCLK, EXTAL and RTCCLK can be used as the clock for counter
    - The division ratio of the clock can be set to 1, 4, 16, 64, 256 and 1024 by software

- PDMA Controller
  - Support up to 8 independent DMA channels
  - Descriptor or No-Descriptor Transfer mode
  - A simple Xburst®-1 CPU supports smart transfer mode controlled by programmable

firmware
- Transfer data units: 1-byte, 2-byte, 4-byte, 16-byte, 32-byte, 64-byte, 128-byte
- Transfer number of data unit: 1 ~ 2^{24} - 1
- Independent source and destination port width: 8-bit, 16-bit, 32-bit
- Fixed three priorities of channel groups: 0~3, highest; 4~11: mid; 12~31: lowest
- An extra INTC IRQ can be bound to one programmable DMA channel

- RTC (Real Time Clock)
  - Need external 32768Hz oscillator for 32KHz clock generation.
  - RTCLK selectable from the oscillator or from the divided clock of EXCLK, so that 32k crystal can be absent if the hibernating mode is not needed
  - 32-bits second counter
  - Programmable and adjustable counter to generate accurate 1 Hz clock
  - Alarm interrupt, 1Hz interrupt
  - Stand alone power supply, work in hibernating mode
  - Power down controller
  - Alarm wakeup
  - External pin wakeup with up to 2s glitch filter

1.2.6 Peripherals

- General-Purpose I/O ports
  - Each port can be configured as an input, an output or an alternate function port
  - Each port can be configured as an interrupt source of low/high level or rising/falling edge triggering. Every interrupt source can be masked independently
  - Each port has an internal pull-up or pull-down resistor connected. The pull-up/down resistor can be disabled
  - GPIO output 4 interrupts, 1 for every group, to INTC

- Three I^2C Controller (SMB0, SMB1, SMB2)
  - Two-wire SMB serial interface – consists of a serial data line (SDA) and a serial clock (SCL)
  - Two speeds
    - Standard mode (100 Kb/s)
    - Fast mode (400 Kb/s)
  - Device clock is identical with pclk
  - Programmable SCL generator
  - Master or slave SMB operation
  - 7-bit addressing/10-bit addressing
  - -level transmit and receive FIFOs
  - Interrupt operation
  - The number of devices that you can connect to the same SMB-bus is limited only by the maximum bus capacitance of 400pF

- One Smart Card Controller (SCC)
  - Supports normal card and UIM card.
  - Supports asynchronous character (T=0) communication modes.
  - Supports asynchronous block (T=1) communication modes.
  - Supports setting of clock-rate conversion factor F (372, 512, 558, etc.), and bit-rate adjustment factor D (1, 2, 4, 8, 16, 32, 12, 20, etc.).
  - Supports extra guard time waiting.
  - Auto-error detection in T=0 receive mode.
  - Auto-character repeat in T=0 transmit mode.
  - Transforms inverted format to regular format and vice versa.
  - Support stop clock function in some power consuming sensitive applications.

- One Synchronous serial interfaces (SSI0)
  - 3 protocols support: National's Microwire, TI's SSP, and Motorola's SPI
  - Full-duplex or transmit-only or receive-only operation
  - Programmable transfer order: MSB first or LSB first
  - Configurable normal transfer mode or Interval transfer mode
  - Programmable clock phase and polarity for Motorola's SSI format
  - Two slave select signal (SSI0_CE0_ / SSI0_CE1_) supporting up to 2 slave devices
  - Back-to-back character transmission/reception mode
  - Loop back mode for testing
  - Data transfer up to 30Mbits/s

- Three UARTs (UART0, UART1, UART2)
  - Full-duplex operation
  - 5-, 6-, 7- or 8-bit characters with optional no parity or even or odd parity and with 1, 1½, or 2 stop bits
  - Independently controlled transmit, receive (data ready or timeout), line status interrupts
  - Internal diagnostic capability Loopback control and break, parity, overrun and framing-error is provided
  - Separate DMA requests for transmit and receive data services in FIFO mode
  - Supports modem flow control by software or hardware
  - Slow infrared asynchronous interface that conforms to IrDA specification
  -
- Two MMC/SD/SDIO controllers (MSC0, MSC1)
  - Fully compatible with the MMC System Specification version 4.5
  - Support SD Specification 3.0
  - Support SD I/O Specification 1.0 with 1 command channel and 4 data channels
  - Consumer Electronics Advanced Transport Architecture (CE-ATA – version 1.1)
  - Maximum data rate is 50MBps
  - Both support MMC data width 1bit ,4bit, only MSC0 support 8bit

- Built-in programmable frequency divider for MMC/SD bus
- Built-in Special Descriptor DMA
- Mask-able hardware interrupt for SDIO interrupt, internal status and FIFO status
- Multi-SD function support including multiple I/O and combined I/O and memory
- IRQ supported enable card to interrupt MMC/SD controller
- Single or multi block access to the card including erase operation
- Stream access to the MMC card
- Supports SDIO read wait, interrupt detection during 1-bit or 4-bit access
- Supports CE-ATA digital protocol commands
- Support Command Completion Signal and interrupt to CPU
- Command Completion Signal disable feature
- The maximum block length is 4096bytes

- USB 2.0 OTG interface
  - Complies with the USB 2.0 standard for high-speed (480 Mbps) functions and with the On-The-Go supplement to the USB 2.0 specification
  - Operates either as the function controller of a high- /full-speed USB peripheral or as the host/peripheral in point-to-point or multi-point communications with other USB functions
  - Supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP)
  - UTMI+ Level 3 Transceiver Interface
  - Soft connect/disconnect
  - 8 endpoints in device mode, 16 channels for host mode.
  - Dedicate FIFO
  - Supports control, interrupt, ISO and bulk transfer

- MAC controller
  - 10/100 Mbps operation
  - Supports RMII PHY interfaces
  - Supports VLAN and CRC
  - Station Management Agent (SMA)
  - remote wake-up frame and magic packet frame processing

- OTP Slave Interface
  - Total 1Kb.

1.2.7 Bootrom

16KB Boot ROM memory

Section 2

CORE FUNCTIONS

2 CPU

Enhanced features of CPU core include:

- Enhanced MXU implements XBurst SIMD instruction set release I and release II
- Full implementation of MIPS32 integer instruction release II
- PMON, processor performance monitor
- FPU, floating point unit implemented to improve floating point number processing ability
- Unified level 2 cache that is transparent for programmer

2.1 Block Diagram

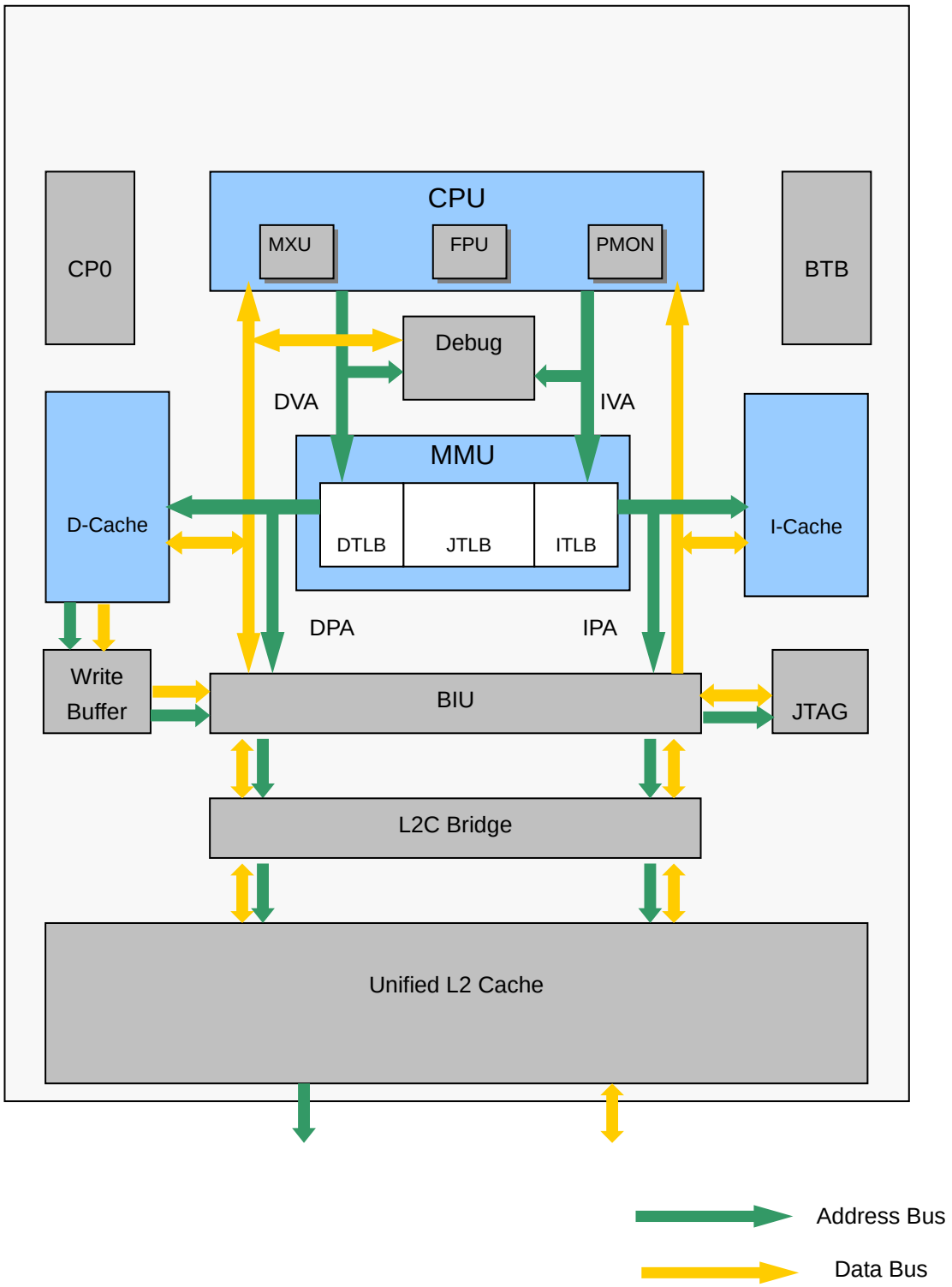


Figure 2-1 Structure of CPU core

2.2 Extra Features of the CPU core

Item	Features
Media Extension Unit (MXU)	● XBurst SIMD instruction set release I and release II ● fully pipelined
Integer Unit with MIPS32 integer instruction release II	● non full pipelined implementation for most of MIP32 integer instruction release II, need 2 ~ 4 interlock cycles
Floating Point Unit (FPU)	● Comply with IEEE754 standard ● Support single and double format ● not fully pipelined implementation
Performance Monitor (PMON)	● Real-time monitor ● Dedicated CP0 interface
Unified Level 2 Cache	● Size: 128K bytes ● 4 way set association with LRU replacement ● Write from Level 1 data cache always write through to memory ● Programmer transparent, that is, those CACHE instructions managing L1 cache can manipulate L2 cache automatically
Processor ID	Value read from CP0.PRId is 0x2ed1024f

Please refer to documents XBurst-ISA and XBurst1_PM for ISA and programming relative details.

2.3 Instruction Cycles

Most instructions have one cycle repeat rate, that is, when the pipeline is fully filled, there is one instruction issued per clock cycle. However, some particular instructions require extra cycles. Following table lists cycle consumption of all instructions belonging to XBurst-ISA implemented.

1st Instruction	2nd Instruction	Cycles	Description
WAIT	**Anyone**	variable	WAIT instruction will be repeatedly executed until an interrupt arises.
MTC0 **TLBWI/TLBWR** **TLBP/TLBR**	**Anyone**	4	3 extra interlock cycles.
CACHE	**Anyone**	2	1 extra interlock cycles.
JMP/BC	**Anyone** **(delay slot)**	4/1	0 cycle penalty when BTB predicts taken and the branch is taken or BTB predicts untaken and the branch is untaken or BTB miss and the branch is untaken. Otherwise, extra 3 cycles penalty.
BCL	**Anyone** **(delay slot)**	5/4/2/1	0 cycle penalty when BTB predicts taken and branch is taken, otherwise: 1 BTB miss, branch is taken, 3 cycles penalty.

			2 BTB miss, branch is untaken, 1 cycle penalty. 3 BTB predict taken, branch is untaken, 4 cycles penalty. 4 BTB predict untaken, branch is taken, 3 cycles penalty.
MULT/MULTU **MADD/MADDU** **MSUB/MSUBU**	**MULT/MULTU** **MADD/MADDU** **MSUB/MSUBU**	4	3 extra interlock cycles due to MDU operating hazard.
	MUL/DIV/DIVU	4	3 extra interlock cycles due to MDU operating hazard.
	MFHI/MFLO **MTHI/MTLO**	4	3 extra interlock cycles due to MDU operating hazard.
	Any other	1	No data dependency or hazards exist.
MUL	**MULT/MULTU** **MADD/MADDU** **MSUB/MSUBU**	4	3 extra interlock cycles due to MDU operating hazard.
	MUL/DIV/DIVU	4	3 extra interlock cycles due to MDU operating hazard.
	MFHI/MFLO **MTHI/MTLO**	4	3 extra interlock cycles due to MDU operating hazard.
	Any other	4/3/2/1	If the second instruction has RAW data dependency, 3 extra interlock cycles; similarly, 2 extra for the third RAW one and 1 extra for the forth RAW one, otherwise, 0 cycle penalty.
DIV/DIVU	**MULT/MULTU** **MADD/MADDU** **MSUB/MSUBU** **MUL/DIV/DIVU**	4~35	3~34 extra interlock cycles determined by characteristic value of divider and dividend.
	MFHI/MFLO	2~34	1~33 interlock cycles determined by characteristic value of divider and dividend.
	Any other	1	No data dependency or hazards exist.
MFHI/MFLO/MFC0	**Anyone**	4/3/2/1	If the second instruction has RAW data dependency, 3 extra interlock cycles, similarly, 2 extra for the third RAW one and 1 extra for the forth RAW one, otherwise, 0 cycle penalty.
LW/LL **LWL/LWR** **LB/LBHU** **LH/HU**	**Anyone**	4/3/2/1	If the second instruction has RAW data dependency, 3 extra interlock cycles, similarly, 2 extra for the third RAW one and 1 extra for the forth RAW one,

LXW **LXH/LXHU** **LXB/LXBU**			otherwise, 0 cycle penalty.
D16MUL/D16MULF **D16MAC/D16MACF** **D16MULE/D16MACE**	**SIMD instruction**	3/2/1	If the second SIMD instruction has RAW data dependency, 2 extra interlock cycles, similarly, 1 extra for the third RAW one, otherwise, 0 cycle penalty.
	Any other	1	No data dependency or hazards exist.
D32ACC/Q16ACC **Q8SAD** **S32MAX/S32MIN** **D16MAX/D16MIN** **D32ACCM/D32ASUM** **Q16ACCM/D16ASUM**	**SIMD instruction**	2/1	If the second SIMD instruction has RAW data dependency, 1 extra interlock cycle, otherwise, 0 cycle penalty.
	Any other	1	No data dependency or hazards exist.
S32LDD/S32LDDV **S32LDI/S32LDIV** **S32LDDR/S32LDDVR** **S32LDIR/S32LDIVR** **S16LDD/S16DI** **S8LDD/S8LDI**	**SIMD instruction**	2/1	If the second SIMD instruction has RAW data dependency, 1extra interlock cycle, otherwise, 0 cycle penalty.
	Any other	1	No data dependency or hazards exist.
S32I2M	**SIMD instruction**	2/1	If the second SIMD instruction has RAW data dependency, 1extra interlock cycle, otherwise, 0 cycle penalty.
	Any other	1	No data dependency or hazards exist.
S32M2I	**Anyone**	4/3/2/1	If the second instruction has RAW data dependency, 3 extra interlock cycles, similarly, 2 extra for the third RAW one and 1 extra for the forth RAW one, otherwise, 0 cycle penalty.
S32EXTR **S32EXTRV**	**SIMD instruction**	2/1	If the second SIMD instruction has RAW data dependency, 1extra interlock cycle, otherwise, 0 cycle penalty.
	Any other	1	No data dependency or hazards exist.
Others	**Anyone**	1	--

NOTE: JMP denotes J and JR instructions; BC denotes branch conditionally instructions; BCL denotes
branch conditionally and likely instructions.

2.4 PMON

PMON is a simple performance monitor. In X1000, PMON can make real-time monitoring for following hardware events.

- I-cache miss times, D-cache miss times
- Total issued instructions, Discarded instructions
- Pipeline freeze cycles, CPU clock cycles
- TLB exceptions caused by instruction fetch, TLB exceptions caused by data load/store

Moreover, in X1000, PMON can be configured to work in expected mode.

- Normal mode (when PMON is enabled, always work until it is disabled)
- User mode (when PMON is enabled, only work in user mode and paused in kernel mode)
- Kernel mode (when PMON is enabled, only work in kernel mode and paused in user mode)
- PC mode (when PMON is enabled, only work when PC locates in preset address range)

A dedicated software interface is devised to manipulate PMON in kernel mode, that is, CP0 Config4 ~ Config7 registers are extended for PMON. Refer to chapter of CP0 in the document XBurst1_PM for detail. However, since new function and bit fields has been expanded for Config7 register, following definition of Config7 is the precise description for X1000.

	Config16																select 7													
Bit	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
	PC_HI																PEVENT				MODE			PME	Reserved	PK	Reserved			ALLOC	BTBV	BTBE
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:16	PC_HI	If PMON work in PC range mode, the valid range is PC_HI,0000 ~ PC_HI,FFFC.	RW
15:12	PEVENT	Event pair encoding. 0000: count of pipeline freeze cycles, count of cpu clock cycles 0001: times count of icache-miss, times count of dcache-miss 0010: count of discarded instructions, count of issued instructions 0011: count of TLB exceptions caused by fetching instruction, count of TLB exceptions caused by data load/store 0100 – 1111: reserved	RW
11:9	MODE	000: null mode, work unconditionally 001: work in user mode 010: work in kernel mode 011: work in specific PC range	RW
8	PME	PMON enable bit. 0: disable; 1: enable.	RW
7	Reserved	Writing has no effect, read as zero.	R

6	PK	Partial kernel mode. 0: forbid; 1: permit. Refer to later description.	RW
5:3	Reserved	Writing has no effect, read as zero.	R
2	ALLOC	Allocate hint of PREF instruction. 0: enabled (default); 1: disabled.	RW
1	BTBV	BTB invalid. Writing 1 to this bit to invalidates BTB.	W
0	BTBE	BTB enable. 0: enabled (default); 1: disabled.	RW

2.4.1 Fundamental

When PMON is enabled (set value 1 to config7.bit8), one preset event pair determined by config7.bit15~bit12 will be continuously monitored until PMON is disabled (set value 0 to config7.bit8).

Finally, loading values of CP0.config4~CP0.config6 can get monitored result.

2.5 Partial Kernel Mode

Setting 1 to config7.bit6 can permit applications in user mode possess some kernel mode oriented resources including CACHE instructions. This is a shortcut for those performance sensitive applications such as video codec. However, OS must make serious control for config7.bit6 and those dedicated resources to forbid this partial-kernel-mode permission for those malicious applications.

Section 3

IMAGE CORE

3 JPEG

3.1 Overview

JPEG module is a jpeg encoding unit in chip X1000.

Features:

- Baseline ISO/IEC 10918-1 JPEG compliant
- 8-bit pixel depth support
- Support for YUY2 ([Y0U0Y1V0]) color
- Up to four programmable Quantization tables
- Fully programmable Huffman tables
- Image size up to 2M pixel

3.2 Register Definition

3.2.1 Task trigger (TRIG)

	TRIG																												0x10008	
Bit	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
	DHA																												Reserved		MD	RUN
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0

Bits	Name	Description	RW
31:4	DHA	Descriptor Head Address. For ACFG mode, the descriptor chain must be located in the external memory and should be 32 word aligned. For TRAN mode, the descriptor chain must be located in the on-chip-memory (TCSM/SRAM) and should be 4 word aligned.	RW
3:2	Reserved	Writing has no effect, read as zero.	R
1	MD	VDMA work mode 0: ACFG mode 1: TRAN mode	RW
0	RUN	VDMA trigger bit, set 1 to startup VDMA.	W

3.2.2 VDMA status (STAT)

	STAT																												0x1000C	
Bit	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
	Reserved																												ACFG_ERR	ACFG_END	DESC_END	Reserved
RST	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

Bits	Name	Description	RW
31:4	Reserved	Writing has no effect, read as zero.	R
3	ACFG_ERR	ACFG error	RW
2	ACFG_END	ACFG end	RW
1	DESC_END	Indicates the ending of final transfer task in TRAN mode	RW
0	Reserved	Writing has no effect, read as zero.	R

3.2.3 Global control information

	GLBC_INFO																												0xE0004	
Bit	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
	P3V		P3H		P2V		P2H		P1V		P1H		P0V		P0H		STRD								reserved		NCOL		DE	RSM	MD	EN
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0

Bits	Name	Description	R/W
31:30	P3V	Component 3 plane vertical size of a MCU Note: 1. its unit is 8x8 block 2. its only be used when GLBC_INFO.MD was set as UNIV mode 3. its only be used when current component was content in the MCU, for example in YUV420 MCU component 3 plane would not be used.	RW
29:28	P3H	Component 3 plane horizontal size of a MCU	RW
27:26	P2V	Component 2 plane vertical size of a MCU	RW
25:24	P2H	Component 2 plane horizontal size of a MCU	RW
23:22	P1V	Component 1 plane vertical size of a MCU	RW
21:20	P1H	Component 1 plane horizontal size of a MCU	RW
19:18	P0V	Component 0 plane vertical size of a MCU	RW

17:16	P0H	Component 0 plane horizontal size of a MCU	RW
15:8	STRD	Component plane stride Note: 1. its unit is byte 2. the LSB 4bits is fixed as 0 (always 16byte aligned) 3. its only be used when GLBC_INFO.MD was set as UNIV mode	RW
7:6	reserved	Writing has no effect, read as zero.	R
5:4	NCOL	Number of color (components) of a MCU minus 1. Example: for YUV420 it should be set as 2	RW
3	WKMD	Work mode for decoder/encoder choose: 0 – work as encoder 1 – work as decoder	RW
2	RSM	Re-Sync-Marker enable	RW
1	DFMD	JPGC data flow mode choose: 0 – SPEC mode 1 – UNIV mode Note: 1. SPEC (specification) mode is suit for tiled YUV420 plane. For SPEC mode, both encoder and decoder would be automatically done without any management during working. 2. UNIV (universal) mode is suit for any other plane formats such as raster YUV444. For UNIV mode, JPGC should be SYNCed and startup MCU by MCU.	RW
0	EN	JPGC module enable Its also be used as the module clock gating for power consumption consideration, when it was reset, JPGC core clock would be gated besides other bit fields of GLBC_INFO such as GLBC_INFO.NCOL. Note: 1. before JPGC startup, it should be set twice to ensure all registers be configured correctly due to the 1st setting is just open the clock.	RW

3.2.4 JPGC trigger

	TRIG																												0xE0000			
Bit	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
	reserved																												TERM	PPT	BST	COPEN
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0

Bits	Name	Description	R/W
31:4	reserved	Writing has no effect, read as zero.	R
3	TERM	Indicates current trigger is the one of the last MCU. Its only need be set in UNIV encoding mode.	RW
2	PPT	Pixel Plane Trigger. In SPEC mode, it should be only set once as the startup of pixel plane working of the whole decoding/encoding process. In UNIV mode, it should be set each MCU as the startup of current MCU's start working.	W
1	BST	Bit-Stream Trigger. For both SPEC and UNIV mode, it should only be set once as the startup of bitstream fetching/storing of the whole decoding/encoding process.	W
0	COPEN	JPGC core open, it should be set at each trigging.	RW

3.2.5 JPGC Status

	STAT																														0x E0008
Bit	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
	ENDF	reserved							BSLEN																							
RST	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31	ENDF	Task done ending flag. For SPEC mode, it indicates whole frame's ending. For UNIV mode, it indicates current MCU's ending.	RW
30:24	reserved	Writing has no effect, read as zero.	R
23:0	BSLEN	Bitstream length (unit: byte) Its only used in encoding process.	RW

3.2.6 Bitstream buffer address

	BSA																														0x E000C
Bit	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
	BSA																									Reserved						
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:7	BSA	Bitstream buffer address. Note: 1. its always be aligned as 32 words 2. it would be updated during the process of decoding/encoding	RW
6:0	reserved	Writing has no effect, read as zero.	R

3.2.7 Component 0 plane buffer address

	P0A																0x E0010													
Bit	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
	P0A																													reserved		
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Bits	Name	Description	R/W
31:3	P0A	Component 0 plane buffer address. Note: 1. In UNIV mode, component plane buffer should be located in the TCSM/SRAM, so the MSB 12bit would not be cared. And also in this mode P0A should be 8-byte aligned. 2. In SPEC mode, component plane buffer should be located in the external memory and should be 256-byte aligned, that is to say the LSB 8bit would not be cared. Furthermore, P0A would be updated during the decoding/encoding process in this mode.	RW
2:0	reserved	Writing has no effect, read as zero.	R

3.2.8 Component 1 plane buffer address

	P1A																0x E0014													
Bit	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
	P1A																													reserved		
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Bits	Name	Description	R/W
31:3	P1A	Component 1 plane buffer address. Note: 1. it is similar with P0A.	RW
2:0	reserved	Writing has no effect, read as zero.	R

3.2.9 Component 2 plane buffer address

	P2A																													0x E0018		
Bit	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
	VPU_BASE												P2A																	reserved		
RST	0	0	0	1	0	0	1	1	0	0	1	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Bits	Name	Description	R/W
31:20	VPU_BASE	Fixed as 0x132	R
19:3	P2A	Component 2 plane buffer address. Note: 1. its only used in UNIV mode	RW
2:0	reserved	Writing has no effect, read as zero.	R

3.2.10 Component 3 plane buffer address

	P2A																													0x E001C		
Bit	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
	VPU_BASE												P3A																	reserved		
RST	0	0	0	1	0	0	1	1	0	0	1	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0

Bits	Name	Description	R/W
31:20	VPU_BASE	Fixed as 0x132	R
19:3	P3A	Component 3 plane buffer address. Note: 1. its only used in UNIV mode	RW
2:0	reserved	Writing has no effect, read as zero.	R

3.2.11 MCU number

	NMCU																													0x E0028	
Bit	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
	reserved						NMCU																									
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:26	reserved	Writing has no effect, read as zero.	R
25:0	NMCU	Number of MCU minus 1 to be processing of current picture	RW

3.2.12 Re-Sync-Marker gap number

	NRSM																													0x E002C	
Bit	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
	reserved																													NRSM		
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:3	reserved	Writing has no effect, read as zero.	R
2:0	NRSM	Number of MCU minus 1 between 2 Re-Sync-Marker.	RW

3.2.13 Component configure information

	C0CI C1CI C2CI C3CI																													0x E0030 0x E0034 0x E0038 0x E003C	
Bit	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
	reserved																								NBLK				QT		HA	HD
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:8	reserved	Writing has no effect, read as zero.	R
7:4	NBLK	Number of block minus 1 in a MCU	RW

3:2	QT	Quant table select of current component	RW
1	HA	Huffman AC table select of current component	RW
0	HD	Huffman DC table select of current component	RW

3.2.14 EFE control

	CTRL																															0x40000
Bit	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
	reserved		YUY2	reserved													CFMT		reserved												EN	RUN	
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:30	reserved	Writing has no effect, read as zero.	R
29	YUY2	When yuv format is YUY2, set it to be 1	RW
28:16	reserved	Writing has no effect, read as zero.	R
15:14	CFMT	Codec format 00: X264; 01: JPEG; 10: VP8E; others: reserved	RW
13:2	reserved	Writing has no effect, read as zero.	R
1	EN	EFE module enable. Its also be used as the module clock gating for power consumption consideration.	RW
0	RUN	EFE startup trigger	W
Note: multi-slice encoding is not supported by EFE, so AUX is needed to do the work (descriptor chain generation and encoder flow management) if multi-slice is used in encoding.			

3.2.15 Encoder slice geometry information

	GEOM																															0x40004
Bit	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
	FSTY								FSTX								LSTY								LSTX							
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	R/W
31:24	FSTY	First MB position of Y direction of current slice	RW
23:16	FSTX	First MB position of X direction of current slice Note: Since only rectangle slice partition of a frame is supported, so FSTX should always be set as 0.	RW

15:8	LSTY	Last MB position of Y direction of current slice	RW
7:0	LSTX	Last MB position of X direction of current slice	RW
Note: in JPEG codec mode, bit27:16 stands for LSTY and bit11:0 stands for LSTX and FSTY/FSTX is always 0			

3.2.16 RAW plane source buffer address

	RAW_SA																															
																	0x40010															
Bit	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
	RAW_SA																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	RAW_SA	RAW plane source buffer address Note: the RAW buffer basement address should be 8-byte aligned	RW

3.2.17 RAW destination buffer basement address

	RAW_DBA																															
																	0x40030															
Bit	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
	VPU_BASE												RAW_DBA											reserved								
RST	0	0	0	1	0	0	1	1	0	0	1	0	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:20	VPU_BASE	Fixed as 0x132	R
19:9	RAW_DBA	RAW destination buffer basement address. EFE fetching RAW data from buffer indicated by RAWY_SA and RAWC_SA then binding and delivering to the buffer indicated by RAW_DBA. It should be 512 byte aligned	RW
8:0	Reserved	Writing has no effect, read as zero.	R

3.2.18 RAW plane stride

	RAW_STR																														0x40038
Bit	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
	RAW_STR																reserved															
RST	0	0	0	1	0	0	1	1	0	0	1	0	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:16	RAW_STR	Raw plane stride	RW
15:0	Reserved	Writing has no effect, read as zero.	R

3.3 Table definition

3.3.1 Huffman Base Table

The Huffman Base Table is used in decoding mode, the space offset of this table is from 0x1200 ~ 0x12FC. More detailed information is defined as the following illustration.

Mapping offset	Description
0x1200 ~ 0x123C	BASE AC 0 value
0x1240 ~ 0x127C	BASE DC 0 value
0x1280 ~ 0x12BC	BASE AC 1 value
0x12C0 ~ 0x12FC	BASE DC 1 value

3.3.2 Huffman MIN Table

The Huffman MIN Table is used in decoding mode, the space offset of this table is from 0x1300 ~ 0x133C. More detailed information is defined as the following illustration.

Mapping offset	Description
0x1300 ~ 0x130C	MIN AC 0 value
0x1310 ~ 0x131C	MIN DC 0 value
0x1320 ~ 0x132C	MIN AC 1 value
0x1330 ~ 0x133C	MIN DC 1 value

3.3.3 Quantization Table

The Quantization Table is used in both decoding and encoding mode, the space offset of this table is from 0x1400 ~ 0x17FC. More detailed information is defined as the following illustration.

Mapping offset	Description
0x1400 ~ 0x14FC	Quantization table 0
0x1500 ~ 0x15FC	Quantization table 1

0x1600 ~ 0x16FC	Quantization table 2
0x1700 ~ 0x17FC	Quantization table 3

3.3.4 Huffman Symbol Table

The Huffman Symbol Table is used in decoding mode, the space offset of this table is from 0x1800 ~ 0x1D3C. More detailed information is defined as the following illustration.

Mapping offset	Description
0x1800 ~ 0x1A84	SYMB AC 0 value
0x1A88 ~ 0x1AB4	SYMB DC 0 and DC 1 value
0x1AB8 ~ 0x1D3C	SYMB AC 1 value

3.3.5 Huffman ENC Table

The Huffman ENC Table is used in encoding mode, the space offset of this table is share with Huffman Symbol Table and it is from 0x1800 ~ 0x1DFC. More detailed information is defined as the following illustration.

Mapping offset	Description
0x1800 ~ 0x1ABC	AC Huffman Table 0
0x1AC0 ~ 0x1D7C	AC Huffman Table 1
0x1D80 ~ 0x1DBC	DC Huffman Table 0
0x1DC0 ~ 0x1DFC	DC Huffman Table 1

Section 4

DISPLAY/CAMERA/AUDIO

4 SLCD Controller

4.1 Overview

The SLCD controller has the capabilities to driving the latest Smart LCD panels. The controller performs the basic memory based frame buffer to smart lcd panel data transfer through usage of a dedicated DMA controller.it support 640x480@60Hz,24BPP.

Features:

- Basic Features
  - Display size up to 640x480@60Hz,24BPP
- Colors Supports
  - Support up to 16,777,216 (16M) colors
- Panel Supports
  - 16bit 8080 once parallel interface
  - 9 bits twice 8080 parallel interface
  - 8 bits twice/third times 8080 parallel
  - 8bit/16bit serial
  - 8bit twice(RG53+GB35)
  - 8bit third times(R8+G8+B8)
  - Supports different size of display panel
  - Supports internal DMA operation and register operation

4.2 Pin Description

Table 4-1 SLCD Pins Description

Name	I/O	Description	Interface
SLCD_DC	Output	Command/Data Select Signal. The polarity of the signal can be programmable.	Serial: DC Parallel: DC
SLCD_WR	Output	Data Sample Signal. The polarity of the signal can be programmable.	Serial: SCLK Parallel: Sample Data with the edge of WR
SLCD_RD	Output	Smart LCD read signal	
SLCD_CE	Output	Smart LCD chip select	
SLCD_TE	Input	Smart LCD tearing effect signal	
SLCD_DAT[*1] [15:0]	Output	The data of SLCD.	Serial: SLCD_DAT [15] Parallel: 16bit SLCD_DAT [15:0] 8bit SLCD_DAT [7:0]

NOTE:

***[1]:** SLCD_DAT [15] is also use as data pin for serial.

4.3 Block Diagram

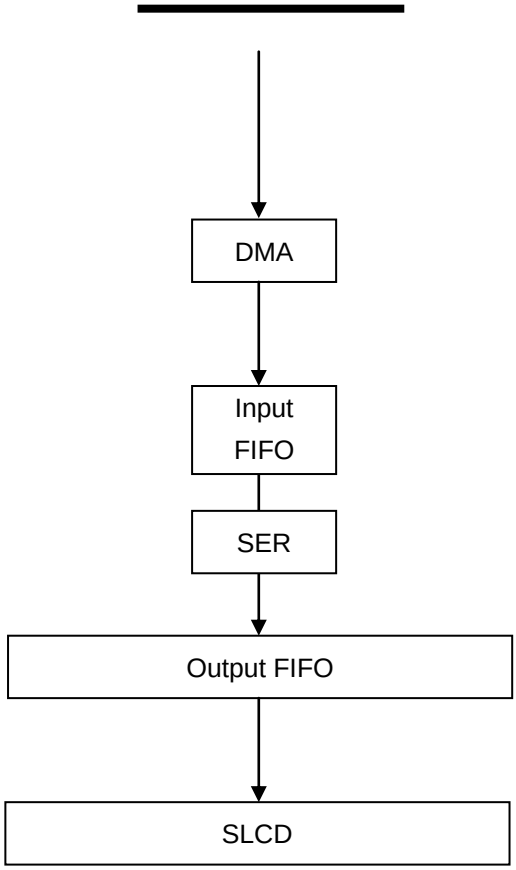


Figure 4-1 Block Diagram of SLCD

4.4 Register Description

Following chapter will descript the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute

   R - Read only

   W - Write only

   RW - read and write

   RCW - read and write, but clear to 0 by read

   RSW - read and write, but set to 1 by read

   RWC - read and write, clear to 0 by write 1, write 0 has no effect

RWS - read and write, set to 1 by write 1, write 0 has no effect

RC - read only, and clear to 0 by read

RS - read only, and set to 1 by read

SPEC - special access method, relate to its description

3. Reset Value

1 - reset to 1

0 - reset to 0

? - value unknown after reset

Table 4-2 Registers Memory Map-Address Base

Name	Base	Description
SLCDC	0x13050000	Address base of SLCDC

Table 4-3 SLCD Controller Registers Description

Name	RW	Description	Address offset	Access Size
LCDCFG	RW	configure register	0x0000	32
LCDCTRL	RW	control register	0x0030	32
LCDSTATE	RW	status register	0x0034	32
LCDOSDC	RW	OSD configure register	0x0100	32
LCDOSDCTRL	RW	OSD control register	0x0104	32
LCDOSDS	RW	OSD STATUS register	0x0108	32
LCDBGC0	RW	background0 color register	0x010C	32
LCDBGC1	RW	background1 color register	0x02C4	32
LCDKEY0	RW	Foreground Color Key Register 0	0x0110	32
LCDKEY1	RW	Foreground Color Key Register 1	0x0114	32
LCDALPHA	RW	ALPHA Register	0x0118	32
LCDRGBC	RW	RGB Control	0x0090	32
LCDVAT	RW	Virtual Area Setting	0x000C	32
LCDDAH	RW	Display Area Horizontal Start/End Point	0x0010	32
LCDDAV	RW	Display Area Vertical Start/End Point	0x0014	32
LCDXYP0	RW	Foreground 0 XY Position Register	0x0120	32
LCDXYP1	RW	Foreground 1 XY	0x0124	32

		Position Register		
LCDSIZE0	RW	Foreground 0 Size Register	0x0128	32
LCDSIZE1	RW	Foreground 1 Size Register	0x012C	32
LCDVSYNC	RW	Vertical Synchronize Register	0x0004	32
LCDHSYNC	RW	Horizontal Synchronize Register	0x0008	32
LCDIID	R	Interrupt ID Register	0x0038	32
LCDDA0	RW	Descriptor0 Address Registers	0x0040	32
LCDSA0	R	Source Address Registers0	0x0044	32
LCDFID0	R	Frame ID Registers0	0x0048	32
LCDCMD0	R	DMA0 Command Registers	0x004C	32
LCDOFFS0	R	DMA0 OFFSIZE Registers	0x0060	32
LCDPW0	R	DMA0 Page Width Registers	0x0064	32
LCDCNUM0/LCDPOS0	R	DMA0 Commend Counter Registers	0x0068	32
LCDDESSIZE0	R	Foreground0 x Size in Descriptor	0x006C	32
LCDDA1*2	RW	Descriptor1 Address Registers	0x0050	32
LCDSA1*2	R	Source Address Registers1	0x0054	32
LCDFID1*2	R	Frame ID Registers1	0x0058	32
LCDCMD1*2	R	DMA1 Command Registers	0x005C	32
LCDOFFS1*2	R	DMA1 OFFSIZE Registers	0x0070	32
LCDPW1*2	R	DMA1 Page Width Registers	0x0074	32
LCDCNUM1/LCDPOS1*2	R	DMA1 Commend Counter Registers	0x0078	32
LCDDESSIZE1*2	R	Foreground1 x Size in Descriptor	0x007C	32
LCDPCFG	RW	Priority level threshold configure	0x02C0	32

		Register		
MCFG	RW	SLCD Configure Register	0x00A0	32
MCFG_NEW	RW	new SLCD Configure Register	0x00B8	32
MCTRL	RW	SLCD Control Register	0x00A4	32
MSTATE	R	SLCD Status Register	0x00A8	32
MDATA	RW	SLCD Data Register	0x00AC	32
WTIME	RW	SLCD Wait Time Register	0x00B0	32
TASH	RW	Address Setup and hold time register	0x00B4	32
SMWT	RW	Wait Time Register	0x00BC	32

4.4.1 Configure Register (LCDCFG)

	LCDCFG																BASE+0x0000															
Bit	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
	Reserved														INVDAT		Reserved															
RST	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

Bits	Name	Description	RW
31:18	Reserved	Write has no effect,read as zero	R
17	INVDAT	Inverse output data. 0: normal; 1: inverse.	RW
16:0	Reserved	Write has no effect,read as zero	R

4.4.2 Control Register (LCDCTRL)

	LCDCTRL																		BASE+0x0030													
Bit	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
	Reserved	BST			Reserved														EOFM	SOFM	Reserved	IFUM0	Reserved		QDM	BEDN	PEDN	Reserved	ENA	BPP0		
RST	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

Bits	Name	Description	RW
31	Reserved	Write has no effect,read as zero	RW
30:28	BST	Burst Length Selection. 000: 4 word 001: 8 word 010: 16 word 011: 32 word 100: 64 word	RW
27:14	Reserved	Write has no effect,read as zero	RW
13	EOFM	Mask end of frame interrupt. 0: INT-disabled; 1: INT-enabled.	RW
12	SOFM	Mask start of frame interrupt. 0: INT-disabled; 1: INT-enabled.	RW
11	Reserved	Write has no effect,read as zero	RW
10	IFUM0	Mask in FIFO 0 under run interrupt. 0: INT-disabled; 1: INT-enabled.	RW
9:8	Reserved	Write has no effect,read as zero	RW
7	QDM	Mask LCD quick disable done interrupt. 0: INT-disabled; 1: INT-enabled.	RW
6	BEDN	Endian selection. 0: same as system Endian; 1: reverse endian format.	RW
5	PEDN	Endian in byte. 0: msb first; 1: lsb first.	RW
4	Reserved	Write has no effect,read as zero	RW
3	ENA	Enable controller. 0: disable; 1: enable.	RW
2:0	BPP0	Bits Per Pixel of foreground0. 000: Reserved 001: Reserved 010: Reserved 011: Reserved 100: 15/16bpp 101: 18bpp/24bpp 110: 24bpp compressed 111: 30bpp	R

4.4.3 Status Register (LCDSTATE)

	LCDSTATE																BASE+0x0034													
Bit	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
	Reserved																								QD	Reserved	EOF	SOF	Reserved	IFU0	Reserved	
RST	0	1	1	1	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

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect, read as zero.	R
7	QD	LCD Quick disable. 0: not been quick disabled; 1: quick disabled done. It can only be set high by hardware, and software can only write 0	RW
6	Reserved	Writing has no effect, read as zero.	R
5	EOF	End of Frame indicate bit. set 1 by hardware,write 0 by software	RW
4	SOF	Start of Frame indicate bit. set 1 by hardware,write 0 by software	RW
3	Reserved	Writing has no effect, read as zero.	R
2	IFU0	In FIFO 0 under run. set 1 by hardware,write 0 by software	RW
1:0	Reserved	Writing has no effect, read as zero.	R

Note: EOF and SOF can only be set high by hardware, software can only write 0

4.4.4 OSD Control Register (LCDOSDCTRL)

	LCDOSDCTRL																BASE+0x0104															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										RGB0	Reserve				
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:6	Reserved	Writing has no effect, read as zero.	R
5	RGB0	Bpp16 RGB mode of foreground0. 0: RGB565; 1: RGB555.	R
4:0	Reserved	Writing has no effect, read as zero.	

4.4.5 Background0 Color Register (LCDBGC0)

	LCDBGC0																BASE+0x010C															
Bit	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
	Reserved		Red[9:0]										Green[9:0]										Blue[9:0]									
RST	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

Bits	Name	Description	RW
31:30	Reserved	Writing has no effect, read as zero.	R
29:20	Red	Red part or Y part of background0.	RW
19:10	Green	Green part or Cb part of background0.	RW
9:0	Blue	Blue part or Cr part of background0.	RW

4.4.6 Display Area Horizontal Start/End Point (LCDDAH)

	LCDDAH																													BASE+0x0010
Bit	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
	Reserved																HDE															
RST	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

Bits	Name	Description	RW
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	HDE	Horizontal display area end. (in dot clock)	RW

4.4.7 Display Area Vertical Start/End Point (LCDDAV)

	LCDDAV																													BASE+0x0014
Bit	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
	Reserved																VDE															
RST	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

Bits	Name	Description	RW
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	VDE	Vertical display area end position. (in line clock)	RW

4.4.8 Foreground 0 XY Position Register (LCDXYP0)

	LCDXYP0																													BASE+0x0120
Bit	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
	Reserved				YPOS												Reserved				XPOS											
RST	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

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	YPOS	The Y position of top-left part for foreground 0.	R
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	XPOS	The X position of top-left part for foreground 0.	R

4.4.9 Foreground 0 Size Register (LCDSIZE0)

	LCDSIZE0																BASE+0x0128															
Bit	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
	Reserved				Height												Reserved				Width											
RST	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

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	Height	The height-1 of foreground 0.	R
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	Width	The width-1 of foreground 0.	R

4.4.10 REV Signal Setting (LCDREV)

	LCDREV																BASE+0x0024															
Bit	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
	Reserved																			REVE												
RST	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

Bits	Name	Description	RW
15:13	Reserved	Writing has no effect, read as zero.	R
12:0	REVE	when run case in FPGA, LCDREV.REVE is used to be pixclk devider and devider = REVE + 1;	RW

4.4.11 Interrupt ID Register (LCDIID)

LCDIID is a read-only register that contains a copy of the Frame ID register (LCDFID) from the

descriptor currently being processed when a start of frame (SOF) or end of frame (EOF) interrupt is generated. LCDIID is written to only when an unmasked interrupt of the above type is signaled and there are no other unmasked interrupts in the SLCD Controller pending. As such, the register is considered to be sticky and will be overwritten only when the signaled interrupt is cleared by writing the SLCD Controller status register.
LCDIID is written with the last channel to reach that state. (i.e. LCDFID of the last channel to reach SOF would be written in LCDIID if SOF interrupts are enabled). Reserved bits must be written with zeros and reads from them must be ignored.

	LCDIID																BASE+0x0038															
Bit	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
	IID																															
RST	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

Bits	Name	Description	RW
31:0	IID	A copy of Frame ID register, which transferred from Descriptor.	R

4.4.12 Descriptor Address Registers (LCDDAx)

Now the slcdc's descriptor is a 8-word block, aligned on 2-word (8-byte, as axi bus is 64bit wide) boundary, in external memory:

WORD [0] contains the physical address for next LCDDAx.
WORD [1] contains the physical address for LCDSAx.
WORD [2] contains the value for LCDFIDx.
WORD [3] contains the value for LCDCMDx.
WORD[4] contains the value for LCDOFFSx
WORD[5] contialns the value for LCDPWx
WORD[6] contians the value for LCDCNUMx when LCDCMD.CMD=1 or the value for LCDCPOSx when LCDCMD.CMD=0.
WORD[7] contains the value for LCDDESSIZEx

Software must write the physical address of the first descriptor to LCDDAx before enabling the SLCD Controller. Once the SLCD Controller is enabled, the first descriptor is read, and all 8 registers are written by the DMAC. The next frame descriptor pointed to by LCDDAx is loaded into the registers for the associated DMA channel after all data for the current descriptor has been transferred.

NOTE: If only one frame buffer is used in external memory, the LCDDAx field (word [0] of the frame descriptor) must point back to itself. That is to say, the value of LCDDAx is the physical address of itself.

Read/write registers LCDDA0, corresponding to DMA channel 0, contain the physical address of the next descriptor in external memory. The DMAC fetches the descriptor at this location after finishing the current descriptor. On reset, the bits in this register are zero. The target address must be aligned to 8-byte boundary. Bits [2:0] of the address must be zero.

	LCDDA0																BASE+0x0040															
Bit	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
	DA0																															
RST	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

Bits	Name	Description	RW
31:0	DA0	Next descriptor physical address. And descriptor structure as following: WORD [0]: next descriptor physical address WORD [1]: the buffer physical address WORD [2]: the buffer ID value (Only for debug) WORD [3] contains the value for LCDCMDx. WORD[4] contains the value for LCDOFFSx WORD[5] contialns the value for LCDPWx WORD[6] contians the value for LCDCNUMx when LCDCMD.CMD=1 or the value for LCDCPOSx when LCDCMD.CMD=0. WORD[7] contains the value for LCDDESSIZEx	RW

4.4.13 Source Address Registers (LCDSA)

Registers LCDSA0 , corresponding to DMA channels 0, contain the physical address of frame buffer in external memory. The address must be aligned on a 4, 8, or 16 word boundary according to register LCDCTRL.BST in old lcdc. But after rewrite of input DMA, the address just need to be aligned by word, If this descriptor is for frame data, LCDSA points to the memory location of the frame buffer. This address is incremented by hardware as the DMAC fetches data from memory. If desired, the Frame ID Register can be used to hold the initial frame source address.

	LCDSA0																BASE+0x0044															
Bit	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
	SA0																															
RST	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

Bits	Name	Description	RW
31:0	SA0	Buffer start address. (Only for driver debug)	R

4.4.14 Frame ID Registers (LCDFIDx)

Registers LCDFID0, corresponding to DMA 0, contain an ID field that describes the current frame. The particular use of this field is up to the software. This ID register is copied to the SLCD Controller Interrupt ID Register when an interrupt occurs.

	LCDFID0																BASE+0x0048															
Bit	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
	FID0																															
RST	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

Bits	Name	Description	RW
31:0	FID0	Frame ID. (Only for debug)	R

4.4.15 DMA Command Registers (LCDCMDx)

	LCDCMD0																BASE+0x004C															
Bit	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
	SOFINT	EOFINT	CMD	Reserved		FRM_EN	Reserved		LEN																							
RST	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

Bits	Name	Description	RW
31	SOFINT	Enable start of frame interrupt. When SOFINT =1, the DMAC sets the start of frame bit (LCDSTATE.SOF) when starting a new frame. The SOF bit is set after a new descriptor is loaded from memory and before the frame data is fetched. In dual-panel mode, LCDSTATE.SOF is set only when both channels reach the start of frame and both frame descriptors have SOFINT set.	R
30	EOFINT	Enable end of frame interrupt. When EOFINT =1, the DMAC sets the end of frame bit (LCDSTATE.EOF) after fetching the last word in the frame buffer. In dual-panel mode, LCDSTATE.EOF is set only when both channels reach the end of frame and both frame descriptors have EOFINT set.	R
29	CMD	It is used to distinguish command and data in lcm mode. And it is only loaded via DMA channel 0. 1: The data is command 0: The data is data	R

28:27	Reserved	Writing has no effect, read as zero.	R
26	FRM_EN	Indicat this frm is enable. 0:disable; 1:enable.	R
28:27	Reserved	Writing has no effect, read as zero.	R
23:0	LEN	The buffer length value. (in WORD) The LEN bit field determines the number of bytes of the buffer size pointed by LCDSAx. LEN = 0 is not valid. DMAC fetch data according to LEN. Each time one or more word(s) been fetched, LEN is decreased automatically. Software can read LEN.	R

4.4.16 DMA OFFSIZE Registers (LCDOFFSx)

	LCDOFFS0																BASE+0x0060															
Bit	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
	Reserved								OFFSIZE																							
RST	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

Bits	Name	Description	RW
23:0	OFFSIZE0	OFFSIZE value for DMA 0. Indicate the offset in word. *please notice that when you need OFFSIZE function, to set this reg to an un-zero value and also need to set LCDPW0 to indicate how much word in one line of this frame.	R

4.4.17 DMA Page Width Registers (LCDPWx)

	LCDPW0																BASE+0x0064, BASE+0x0074															
Bit	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
	Reserved								PAGEWIDTH																							
RST	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

Bits	Name	Description	RW
23:0	PAGEWIDTH0	Page width for DMA 0,1. * When you set LCDOFFS.OFFSIZE0 to 0, you need keep the PAGEWIDTH0 0.	R

Note: PAGEWIDTH shoule be words in block line that dma need to read in frame buffer. for example,surppose there is a 90 line picture in frame buffer, if you set PAGEWIDTH as large as 30lines, the picture will be divided to 3 blocks of 30lines. Then OFFSIZE indcates the size between 2 pagewidth. But usually we will set PAGEWIDTH as large as 1line, and divide the

picture to 90 blocks, that is to say one block equals one line, and OFFSIZE be 1 line that is interlace.

4.4.18 DMA Commend Counter Registers (LCDCNUMx)

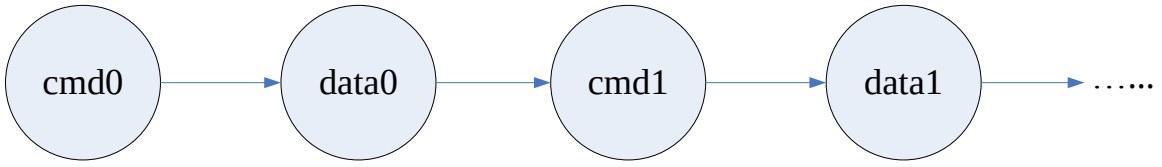
When LCDCMD.CMD = 1, **BASE+0x006** is use as LCDCNUM0 are not used now, set it to 0.

	LCDCNUM0																BASE+0x0068,													
Bit	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
	Reserved																								CNUM							
RST	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

Bits	Name	Description	RW
7:0	CNUM0	Commands' number in this frame transfer by DMA	R

Note1: LCDCNUM should matces the LCDCMD.LEN: when cwidth is 8bits, LCDCNUM=4*LEN, when cwidth is 9bits or 16bits, LCDCNUM=2*LEN, when cwidth is 18bit or 24bits, LCDCNUM=LEN.
Note2:onlye dma0's descrptor can have set LCDCMD.CMD to be 1'b0, and dma's descrptor is as forllows when use slcd:

For dma0



4.4.19 DMA Commend Counter Registers (LCDCPOSx)

When LCDCMD.CMD = 0, **BASE+0x0068** is use as LCDPOS0

	LCDPOS0																BASE+0x0068													
Bit	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
	Reserved	RGB0	BPP0			Reserved			YPOS												XPOS											
RST	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

Bits	Name	Description	RW
31	Reserved	Writing has no effect, read as zero.	R
30	RGB0	Bpp16 RGB mode of foreground0. 0: RGB565; 1: RGB555.	R
29:27	BPP0	Bits Per Pixel of DMA	R

	Bits Per Pixel
000	Reserved
001	Reserved
010	Reserved
011	Reserved
100	15/16bpp
101	18bpp/24bpp
110	24bpp compressed
111	30bpp

26:24	Reserved	Writing has no effect, read as zero.	R
23:12	YPOS0	The Y position of top-left part for foreground 0	R
11:0	XPOS0	The X position of top-left part for foreground 0	R

Note: only low 8 bits of LCDCPOS is readable when LCDCMD.CMD = 0.

4.4.20 Foreground x Size in Descriptor (LCDDESSIZEx)

When LCDCMD.CMD = 0, **BASE+0x006C** is use as LCDDESSIZE0, 1, to indicator the next frame foreground0, 1's size.

	LCDDESSIZE0																BASE+0x006C													
Bit	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
	Reserved								Height												Width											
RST	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

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:12	Height	The height-1 of foreground 0	R
11:0	Width	The width-1 of foreground 0	R

4.4.21 Priority level threshold configure Register (LCDPCFG)

	LCDPCFG																BASE+0x02C0													
Bit	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

	Lcd_pri_md	HP_BST			Reserved	Pcfg2									Pcfg1									Pcfg0								
RST	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The 3 thresholds cut the output fifo into 4 space,. When the entries remained triggers one of them, the priority level will be altered by hardware. And the 3 thresholds must follow the order: pcfg2 ≥ pcfg1 ≥ pcfg0.

Priority level

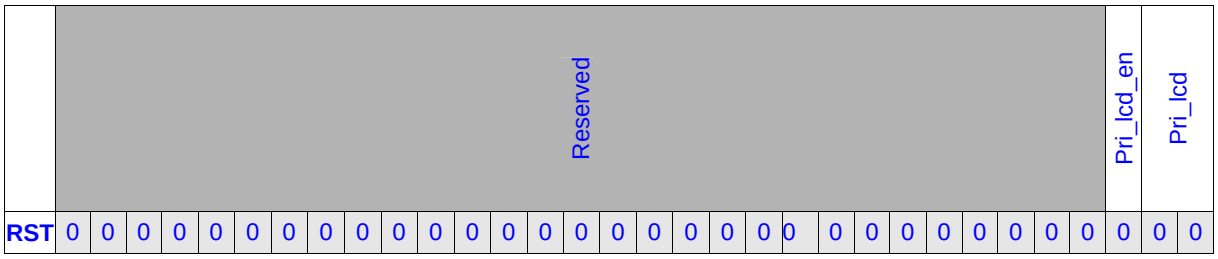
Priority_lv	Space
11	Empty ~ threshold0
10	Threshold0 ~ threshold1
01	Threshold1 ~ threshold2
00	Threshold2 ~ full

Bits	Name	Description	RW
31	Lcd_pri_md	Lcd priority mode. 0: use lcd dynamitic priority level; 1: use arbiter priority level. this bit is useless now	RW
30:28	HP_BST	Highest priority Burst Length Selection.	RW
27	Reserved	Writing has no effect, read as zero.	R
26:18	Pcfg2	Threshold2: 0~511.	RW
17:9	Pcfg1	Threshold1: 0~511.	RW
8:0	Pcfg0	Threshold0: 0~511	RW

	Burst Length
000	4 word
001	8 word
010	16 word
011	32 word
101	Contiue16
100	64 word
111	Disable

4.4.22 LCD Arbiter Priority(LCD_PCFG_ARB)

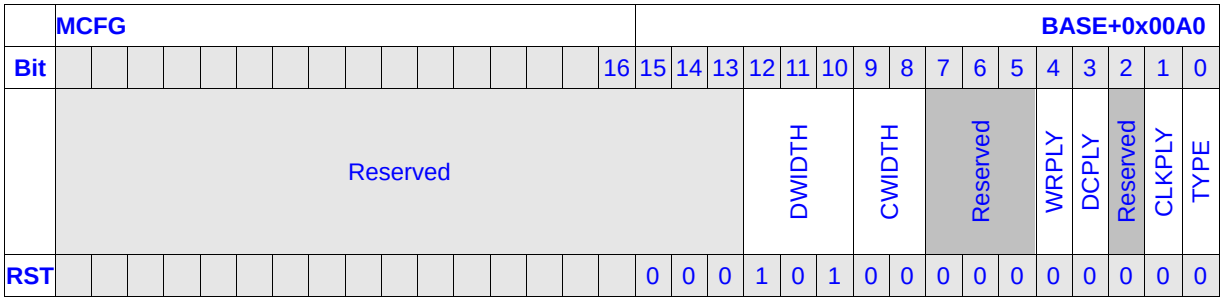
	LCD_PCFG_ARB																															**0x130502C8**
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0



Bits	Name	Description	RW
31:3	Reserved	Writing has no effect, read as zero.	R
2	Pri_lcd_en	1:fix the priority of lcdc0/1 in lcd internal arbiter; 0: use priority of lcdc0/1 generated by lcd in lcd internal arbiter	RW
1:0	Pri_lcd	Set lcdc0/1 priority, just active when pri_lcd_en is 1	RW

4.4.23 SLCD Configure Register (MCFG)

The register MCFG is used to configure SLCD.



Bits	Name	Description	RW
31:13	Reserved	Writing has no effect, read as zero.	R
12:10	DWIDTH[*1]	Do not change these bits	R
9:8	CWIDTH	Command Width. CWIDTH: 00 = 16-bit / 9bit; 01 = 8-bit; 10 = 18-bit; 11 = 24-bit	RW
7:5	Reserved	Writing has no effect, read as zero.	R
4	WRPLY	Do not change this bit	R
3	DCPLY	Do not change this bit	R
2	Reserved	Writing has no effect, read as zero.	R
1	CLKPLY	Do not change this bit	R
0	TTYPE	Do not change this bit	R

NOTE : MCFG.CWIDTH together with MCFG_NEW.CMD_9BIT decide cmd width. other bits of MCFG are useless.

4.4.24 SLCD Configure New Register (MCFG_NEW)

The register MCFG_NEW is used to configure new SLCD.it works together with MCFG.

	MCFG_NEW																BASE+0x00B8													
Bit	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
	Reserved																DWIDTH_NEW			Reserved	6800_md	Cmd_9bit	DTIMES_NEW		Reserved		CSPLY_NEW	RSPLY_NEW	CLKPLY_NEW	DTYPE_NEW	CTYPE_NEW	FMT_CONV
RST														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:13	DWIDTH_NEW	Data Width.it's also useful in MIPI DSI command mode DWIDTH: 3'b000 — Data Width: 8 bit 3'b001 — 9bit 3'b010 — 16bit 3'b011 — 18bit 3'b100 — 24bit Others — Reserved	RW
12	Reserved	Writing has no effect, read as zero.	R
11	6800_md	When high, wr will be inversed and that's 6800's timing Keep low when not in 6800 mode	RW
10	cmd_9bit	Useful when MCFG.CWIDTH is 2'b00 1'b1:9 bit mode 1'b0:16 bit mode	RW
9:8	DTIMES_NEW	Number of pixel clock cycles be used to transmit one pixel with 8bits width data port. 2'b00 : 1-time transfer is used to transmit 1 pixel data, such as sending 8-bit color depth information(RGB332) , one pixel per cycle. 2'b01 : 2-times transfer is used to transmit 1 pixel data, such as sending 16-bit color depth information(RGB565) , one and half	RW

		sub-pixel per cycle. 2'b10 : 3-times transfer is used to transmit 1 pixel data, such as sending 18-bit color depth information(RGB888) , one sub-pixel per cycle. 2'b11 : reserved	
7:6	Reserved	Writing has no effect, read as zero.	R
5	CSPLY_NEW	1'b0:WR will be high when in IDLE(8080 or serial) 1'b1:WR will be low when in IDLE(6800s)	RW
4	RSPLY_NEW	DC Polarity. 0: Command DC = 0, Data DC = 1 1: Command DC = 1, Data DC = 0	RW
3	CLKPLY_NEW	LCD_CLK Polarity. 0: Active edge is Falling 1: Active edge is Rising	
2	DTYPE_NEW	Data Transfer Type. 0: Parallel 1: Serial	RW
1	CTYPE_NEW	Cmd Transfer Type. 0: Parallel 1: Serial	RW
0	FMT_CONV	Format conversion or not 1'b1:use format conversion 1'b0:not use format conversion	RW

NOTE1:
When WIDTH is 8bit and TYPE is serial, TIMES should be zero.
NOTE2:
when send data(include dma mode and register mode) and FMT_CONV is low,or and when send cmd (include dma mode and register mode), slcd will directly give out the low 8/16/18 bits from output fifo directly; when if this bit is set high, new slcd will convert the 24bpp data in output fifo to relative bpps as follows

24bpp data in output fifo:

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
X	X	X	X	X	X	X	X	R7	R6	R5	R4	R3	R2	R1	R0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

8bpp output data(R3G3B20:

								23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
								0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	0	0	0	0	0	0	0	R7	R6	R5	G7	G6	G5	B7	B6

15bpp output data(R5G5B5)

								23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
								0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	R7	R6	R5	R4	R3	G7	G6	G5	G4	G3	B7	B6	B5	B4	B3

16bpp output data(R5G6B5)

								23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
								0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R7	R6	R5	R4	R3	G7	G6	G5	G4	G3	G2	B7	B6	B5	B4	B3

18bpp output data(R6G6B6)

								23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
								0	0	0	0	0	0	R7	R6
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R5	R4	R3	R2	G7	G6	G5	G4	G3	G2	B7	B6	B5	B4	B3	B2

9bit twice:

it’s data format is the same with 18bpp, but the data will be given out in twice,the first time is

								23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
								0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	0	0	0	0	0	0	R7	R6	R5	R4	R3	R2	G7	G6	G5

the second time is

								23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
								0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	0	0	0	0	0	0	G4	G3	G2	B7	B6	B5	B4	B3	B2

Note4:

when send cmd, MCFG_NEW.FMT_CONV is useless, the output cmd is only decided by cmd width. if cmd is 8bits width,only the low 8 bits of slcd data is useful, and one word in buffer will be divided into 4 cmd, the 1st cmd will be data[7:0], second data[15:8], third data[23:16], and the last is data[31:24]; if cmd is 9bits width, only the low 9bit of slcd data is useful, and one word in buffer will be divide into 2 cmd, the 1st cmd will be data[8:0], and the second will be data[24:16], other bits will be lost.; if cmd is 16bits width, only the low 16bit of slcd data is useful, and one word in buffer will be divide into 2 cmd, the 1st cmd will be data[15:0], and the second will be data[31:16]; if cmd width is 18bits or 24bits width, one word in buffer will only send one cmd, only low 18 bits or the total 24bits of slcd data is useful, and the cmd will be data[17:0] or data[23:0].

Note5：

DTIMES_NEW and CTIMES_NEW is only useful for descrptor mode, when in regster mode,

write register once will only send message once. FMT_CONV is only useful for data(include register mode and descriptor mode), when send cmd, send out the relative low bit directly.

4.4.25 SLCD Control Register (MCTRL)

MCTRL is SLCD Control Register.

	MCTRL																				BASE+0x00A4											
Bit																					11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																					NARROW_TE	TE_INV	NOT_USE_TE	DCSI_SEL	MIPI_SLCD	Reserved	FAST_MODE	gate_mask	DMAMODE	DMASTART	Dmatxen
RST																					0	0	0	1	0	0	1	1	1	0	0	0

Bits	Name	Description	RW
31:11	Reserved	Writing has no effect, read as zero.	R
10	NARROW_TE	1'b1: support narrow TE 1'b0:only support TE as wide as 3 pixclk	RW
9	TE_INV	1'b0: te is active high 1'b1:te is active low	RW
8	NOT_USE_TE	1'b1:not wait for TE's relative edge 1'b0: wait for TE's relative edge	RW
7	DCSI_SEL	DSI or CSI sel, it's only used in fpga for debug	RW
6	MIPI_SLCD	1'b1:use mipi slcd (for mipi dsi command mode) 1'b0:not use mipi slcd	RW
5	Reserved	Write has no effect read as zero	R
4	FAST_MODE	1'b1:MCU's frame frequency is faster than smart lcd panel 1'b0: MCU's frame frequency is lower than smart lcd panel	RW
3	GATE_MASK	1'b0:slcd's clk will never be gated 1'b1: slcd's clk (include hclk and pixclk) will be gated when in dma singel mode and not busy. keep this bit 0 when in register mode	RW
2	DMAMODE	SLCD descriptor DMA mode select. 0: DMA will continually transfer data follow descriptor chain 1: DMA will stop when one descriptor finished	RW
1	DMASTART	Only use when DMAMODE = 1, set 1 to restart DMA transfer.	RW
0	DMATXEN	DMA mode enable.	RW

Note: when you clear DMATXEN to use register mode, make sure DMAMODE is set high at the same time. after register mode is finished, you can set DMAMODE high or low depending on wether you want to use dma single mode or dma continue mode.

Note: when use register mode, keep ctimes and dtimes 0,and locate the data or cmd in the relative low bits.

4.4.26 SLCD Status Register (MSTATE)

The register of MSTATE is SLCD status register.

	MSTATE																BASE+0x00A8													
Bit																								7	6	5	4	3	2	1	0
	LCD_ID																Reserved														BUSY
RST																								0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:16	**LCD_ID**	**It's used to point the version of LCDC,in current version, it's 1**	
15:1	Reserved	Writing has no effect, read as zero.	R
0	BUSY	Transfer is working or not. This bit will be set to 1 when transfer is working. It will be cleared by hardware when transfer is finished. 0: not busy 1: busy	R

4.4.27 SLCD Data Register (MDATA)

The register MDATA is used to send command or data to SLCD. When PTR=2'b00, the low 24-bit is used as data. When PTR=2'b01, the low 24-bit is used as command. When PTR=2'b10, the low 24-bit is used as wait time.

	MDATA																												BASE+0x00AC							
Bit	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
	PTR		Reserved						DATA / CMD																							
RST	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

Bits	Name	Description	RW
31:30	PTR	The PTR bit of data register is used to decide the meanings of the low 24-bit. 2'b00: data 2'b01: command	RW

		2'b10: Rserved 2'b11: Reserved	
29:24	Reserved	Writing has no effect, read as zero.	R
23:0	DATA/CMD	Data or Command Register or wait time	RW

4.4.28 SLCD Wait Time Register (WTIME)

	WTIME																BASE+0x00B0													
Bit	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
	DHTIME								DLTIME								CHTIME								CLTIME							
RST	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

DHTIME:high time period when writing data

DLTIME:low time period when writing data

CHTIME:high time period when writing cmd

CLTIME:low time period when writing cmds

The high or low time is described in 8080 mode, it's opposite when in 6800 mode

4.4.29 Address Setup and Hold Time Register (TASH)

	WTIME																BASE+0x00B4													
Bit	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
	Reserved																TAH								TAS							
RST	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

TAS：address setup time

TAH：address hold time

Note：TAS is usefull both in parallel and serial mode, while TAH is only usefull in serial mode.

4.4.30 Slow Mode Wait Time Register(SMWT)

	SLOW_TIME																BASE+0x00BC													
Bit	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
	Reserved																SLOW_TIME															
RST	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

SLOW_TIME: if use the function of slcd to wait for TE's relative edge, slcd will wait for SLOW_TIME's period of pixclk before it begin to flush out data

4.5 SLCD Controller Operation

4.5.1 Set SLCD Controller AHB Clock and Pixel Clock

The SLCD Controller has 2 clock input: AHB clock and pixel clock. The both clocks are generated by CPM (Clock and Power Manager). The frequency of the 2 clocks can be set by CPM registers. lcdc ‘s AHB clock is equal to AHB0 clock (HCLK in CPM spec), and CPM.LPCDR set SLCD pixel clock division ratio. Please refer to CPM spec for detail.

SLCD AHB clock is the SLCD controller’s internal clock while SLCD pixel clock is output to drive SLCD panel. AS for SLCD Panel, the frequency of SLCD AHB clock must be at least 1.5 times of pixclk. SLCD panel determines the frequency of SLCD pixel clock.

4.5.2 Enabling the Controller

If the SLCD controller is being enabled for the first time after system reset or sleep reset, all of the SLCD registers must be programmed as follows:

1 Write the frame descriptors to memory.
2 Program the entire LCD configuration registers except the Frame Descriptor Address Registers (LCDDAx) and the SLCD Controller enable bit (LCDCTRL.ENA).
3 Program LCDDAx with the memory address of the frame descriptor.
4 Enable the SLCD Controller by writing to LCDCTRL.ENA.

If the SLCD Controller is being re-enabled, there has not been a reset since the last programming; only the registers LCDDAx and LCDCTRL.ENA need to be reprogrammed. The SLCD Controller Status Register (LCDSTATE) must also be written to clear any old status flags before re-enabling the SLCD Controller.

Once the SLCD Controller has been enabled, do not write new values to LCD registers except LCDCTRL.ENA.

4.5.3 Disabling the Controller

The SLCD controller can only use quick disable.

1 Quick disabling.
Quick disabling is accomplished by clearing the enable bit, LCDCTRL.ENA. The SLCD Controller will finish any current DMA transfer, stop driving the panel, setting the LCD Quick Disable bit (LCDSTATE.QD) and shut down immediately. This method is intended for situations such as a battery fault, where system bus traffic has to be minimized immediately so the processor can have enough time to store critical data to memory before the loss of power. The SLCD Controller must not be re-enabled until the QD bit is set, indicating that the quick shutdown is complete. Do not set the DIS bit when a quick disabling command has been issued.

NOTE: It is strongly recommended that software set the “LCD Module Stop Bit” in CPM to shut down LCDC clock supply to save power consumption after disable LCDC. Please refer to CPM for detailed information.

4.5.4 Resetting the Controller

At reset, the SLCD Controller is disabled. All SLCD Controller Registers are reset to the conditions shown in the register descriptions.

4.5.5 Frame Buffer

The starting address of frame buffer stored in external memory must be aligned to 4, 8 or 16 words boundary according to register LCDCTRL.BST. The length of buffer must be multiple of word (32-bit).

If LCDCTRL .BST = 0, align frame buffer to 16 word boundary
If LCDCTRL .BST = 1, align frame buffer to 8 word boundary
If LCDCTRL .BST = 2, align frame buffer to 4 word boundary
but as DMA is rewrited, Frame buffer just need to be aligned to word.

One frame buffer contains encoded pixel data of multiple of screen lines; each line of encoded pixel data must be aligned to word boundary. If the length of a line is not the multiple of word, extra bits must be applied to reach a word boundary. It is suggested that the extra bits to be set zero.

4.6 System Memory Format

4.6.1 Data format

1 16bpp

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

2 18bpp

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
0	0	0	0	0	0	0	0	R5	R4	R3	R2	R1	R0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
G5	G4	G3	G2	G1	G0	0	0	B5	B4	B3	B2	B1	B0	0	0

3 24bpp

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	R7	R6	R5	R4	R3	R2	R1	R0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

4 30bpp

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	R9	R8	R7	R6	R5	R4	R3	R2	R1	R0	G9	G8	G7	G6
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
G5	G4	G3	G2	G1	G0	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

5 16bpp with alpha

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A7	A6	A5	A4	A3	A2	A1	A0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R5	R4	R3	R2	R1	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1

6 18bpp with alpha

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A7	A6	A5	A4	A3	A2	A1	A0	R5	R4	R3	R2	R1	R0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
G5	G4	G3	G2	G1	G0	0	0	B5	B4	B3	B2	B1	B0	0	0

7 24bpp with alpha

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A7	A6	A5	A4	A3	A2	A1	A0	R7	R6	R5	R4	R3	R2	R1	R0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

8 24bpp compressed

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLUE 1 [7:0]								RED 0 [7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GREEN 0 [7:0]								BLUE 0 [7:0]							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GLEEN 2 [7:0]								BLUE 2 [7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RED 1 [7:0]								GLEEN 1 [7:0]							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RED 3 [7:0]								GLEEN 3 [7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE3 [7:0]								RED2 [7:0]							

4.6.2 Command Format

1 18-bit command

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
X	X	X	X	X	X	X	X	X	X	X	X	X	X	C17	C16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

2 16-bit command

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

3 9-bit command once

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
X	X	X	X	X	X	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
X	X	X	X	X	X	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

4 8-bit command once

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
C7	C6	C5	C4	C3	C2	C1	C0	C7	C6	C5	C4	C3	C2	C1	C0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C7	C6	C5	C4	C3	C2	C1	C0	C7	C6	C5	C4	C3	C2	C1	C0

5 8-bit command twice (Command = command part + data part)

*Please notice that when you use this kind command, set CWIDTH as 8bit once and set the LCDCNUM.CNUM as doubled the real command number.

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
D7	D6	D5	D4	D3	D2	D1	D0	C7	C6	C5	C4	C3	C2	C1	C0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
D7	D6	D5	D4	D3	D2	D1	D0	C7	C6	C5	C4	C3	C2	C1	C0

4.7 Transfer Mode

Two transfer modes can be used: DMA Transfer Mode and Data Register Transfer Mode. In DMA mode, always transfer commands by DMA 0.

4.7.1 DMA Transfer Mode

Command and data can be recognized by DC bit coming from memory. The format of DMA transfer can be as follows:

1 Command and Data

C	D	D	C	C	D	D	D	C	C	C	D	D	D	D

*Please notice that the command only can insert between two complete frame and the number of command is 0~255.

2 Only Data

D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

*You can also not use command but you still need to use a command descriptor and set the CNUM = 0.

Because DMA transfer mode only can work in OSD mode, you need to configure the panel according OSD mode:

1 Set Color.
* LCDBGC0,1, LCDKEY0, LCDKEY1, LCDALHPA

2 Set Display.
* LCDVAT, LCDDAH, LCDDAV
* LCDVSYNC, LCDHSYNC

3 Set Descriptors.

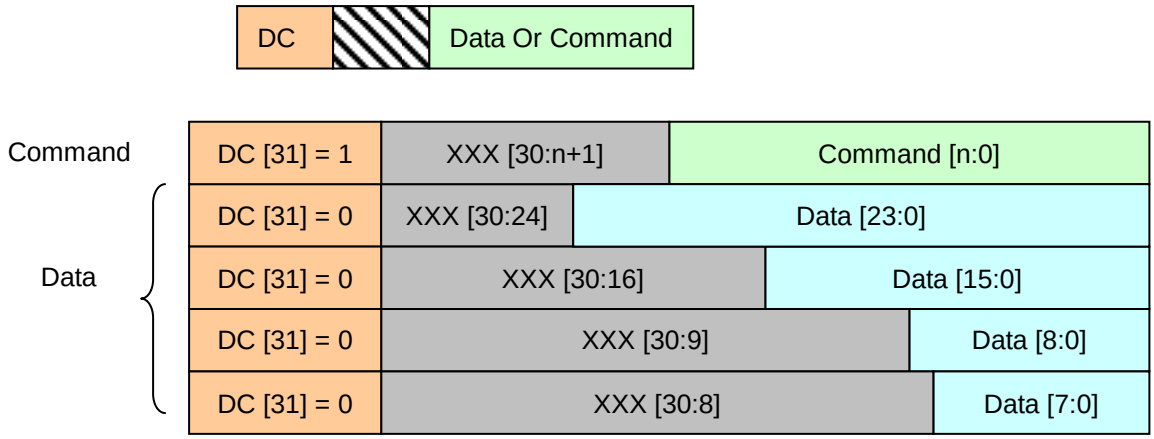
* LCDIID
* LCDDA0, LCDSA0, LCDFID0, LCDCMD0, LCDOFFS0, LCDPW0, LCDCNUM0/LCDPOS0, LCDDESSIZE0
* LCDDA1, LCDSA1, LCDFID1, LCDCMD1, LCDOFFS1, LCDPW1, LCDCNUM1/LCDPOS1, LCDDESSIZE1

4 Enable slcd DMA.

5 Enable LCDC.

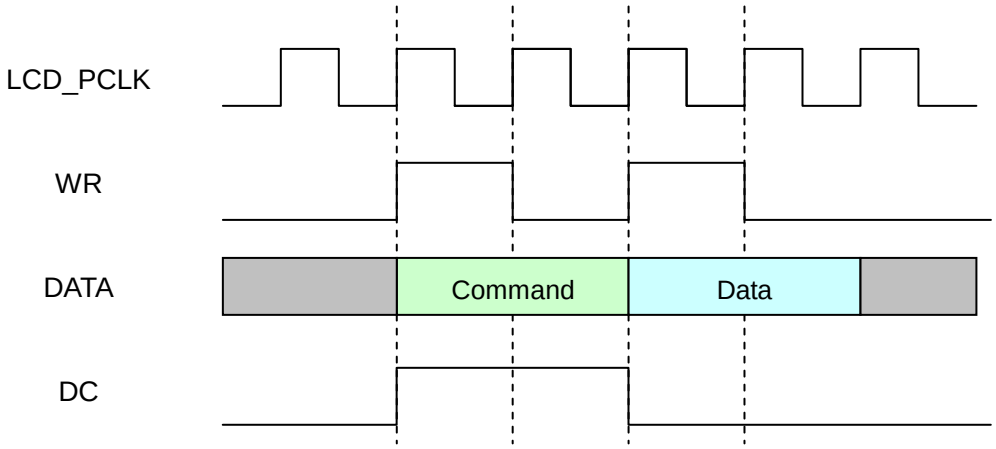
4.7.2 Register Transfer Mode

Each time you can write a command or a data to the register, then it will transfer the DC signal and data or command to SLCD. Command and data can be recognized by DC bit coming from data register. The format of data register transfer can be as follows:

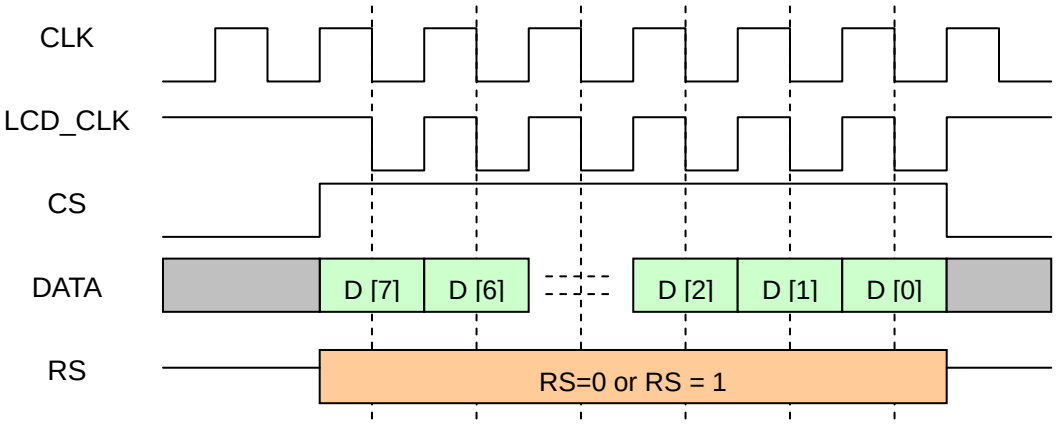


4.8 Timing

4.8.1 Parallel Timing



4.8.2 Serial Timing



4.9 Operation Guide

4.9.1 DMA Operation

1.Start DMA

a Set LCDCFG.MODE to 1101 to choose LCM.

b Set LCDCTRL.BST to choose burst length for transferring.

c Set register LCDIID0, LCDDA0, LCDSA0, LCDFID0, LCDCMD0, LCDOFFS0, LCDPW0, LCDCNUM0, LCDDESSIZE0 to initial internal DMA.

d Also set register LCDIID1, LCDDA1, LCDSA1, LCDFID1, LCDCMD1, LCDOFFS1, LCDPW1, LCDCNUM1, LCDDESSIZE1 when use DMA channel 1 in OSD mode.

e Set MCFG to configure SLCDC.and if use new slcd, you need also configure MCFG_NEW, MCTRL, WTIME,TAS and SMWT

f Before starting DMA, Wait for MSTATE.BUSY == 0.

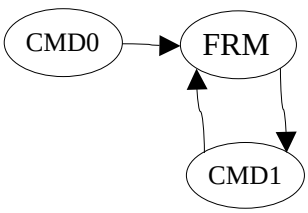
g Set MCTRL.DMATXEN to 1 to prepare DMA transfer.
Note that if you don't want to stop DMA transfer, you need not to check MSTATE.BUSY.

h Set LCDCCTRL.ENA to 1 to start LCDC internal DMA.

i The LCDC internal DMA will transfer data to SLCDC, and SLCDC transfer data to LCM.
Repeat this step till you want to close the SLCDC to transfer data to LCM Panel.

*please notice that use and only use DMA0 to transfer command no matter use DMA0 to transfer frame data or not.

One recommend descriptor chain (CMD0 with CNUM>0 and CMD1 with CNUM=0):



2 Stop DMA transfer.

a Set LCDCCTRL.ENA to 0 to stop LCDC internal DMA at once.

b Wait till MSTATE.BUSY is set to 0 by hardware.
MSTATE.BUSY == 1: there is data in the FIFO waited for transferring to LCM.
MSTATE.BUSY == 0: all data in the FIFO have finished transferring to LCM.

c Set MCTRL.DMATXEN to 0 to stop DMA transfer.

3 Restart DMA transfer.

When MCTRL.DMATXEN is set to 0, and then you want to restart DMA transfer at once, you should ensure that MCTRL.DMATXEN must keep 0 at least three cycles of PIXCLK.

4.9.2 Register Operation

1 Set MCFG, MCFG_NEW, MCTRL, WTIME, TAS, SMWT to configure SLCD.
2 Wait for MSTATE.BUSY == 0.
3 Set MDATA register.
4 Wait for MSTATE.BUSY == 0.
5 Set MDATA register.
6 Wait for MSTATE.BUSY == 0.

recommended steps to use slcd

1. setup LCDC
2. setup SLCD for register mode
3. enable LCDC, write MDATA to initiate smart lcd
4. quick disable LCDC and clear LCDSTATE.QD
5. setup SLCD for descrptor mode, here to make sure the 1st descriptor is cmd and CMDNUM>0, then the descriptor should switch between data and cmd

5 Camera Interface Module

5.1 Overview

The camera interface module (CIM) supports commonly available CMOS or CCD type image sensors. The CIM sources the digital image stream through a common 8-bit parallel digital protocol. The CIM can directly connect to external CMOS image sensors and ITU656 standard video decoders.

5.1.1 Features

- Input image size up to 2M pixels
- Integrated DMA
- Supported data format: YCbCr 4:2:2
- Supports ITU656 (YCbCr 4:2:2) input
- Configurable VSYNC and HSYNC signals: active high/low
- Configurable PCLK: active edge rising/falling
- 128x66 image data receive FIFO (RXFIFO)
- PCLK max. 80MHz
- Configurable output order

5.1.2 Pin Description

Table 5-1 Camera Interface Pins Description

Name	I/O	Width	Description
MCLK	O	1	CIM work clock
PCLK	I	1	Pixel clock from Image Sensor
VSYNC	I	1	Vertical synchronous from Image Sensor
HSYNC	I	1	Horizontal synchronous from Image Sensor
DATA	I	8	Data bus from Image Sensor

5.2 CIM Special Register

The base address of CIM controller is 0x13060000.

Table 5-2 CIM Registers

Name	RW	Reset Value	Offset	Access Size
CIMCFG	RW	0x00000000	0x0000	32
CIMCR	RW	0x00000000	0x0004	32
CIMST	RW	0x00150002	0x0008	32
CIMIID	R	0x00000000	0x000C	32
CIMDA	RW	0x00000000	0x0020	32

CIMFA	R	0x00000000	0x0024	32
CIMFID	R	0x00000000	0x0028	32
CIMCMD	R	0x00000000	0x002C	32
CIMWSIZE	RW	0x00000000	0x0030	32
CIMWOFFSET	RW	0x00000000	0x0034	32
CIMCR2	RW	0x00000000	0x0050	32
CIMFS	RW	0x00000000	0x0054	32
CIMIMR	RW	0x00000F0D	0x0058	32

5.2.1 CIM Configuration Register (CIMCFG)

	CIMCFG																														0x0000
Bit	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
	EEOFEN	Reserved							BS3		BS2		BS1		BS0		INV_DAT	VSP	HSP	PCP	BURST_TYPE		DUMMY	E_VSYNC	LM	PACK			FP	E_HSYNC	DSM	
RST	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

Bits	Name	Description	RW
31	EEOFEN	Early EOF Mode Enable. 0: EEOF mode is disabled 1: When CIMCR.EEOF_LINE lines data has been transferred of a frame, the EEOF flag will be set, and the EEOF interrupt will occur	R W
30:24	Reserved	Writing has no effect. Read as zero.	R
23:22	BS3	Byte Selection. Using IByte0 stands for the 1st byte received IByte1 stands for the 2nd byte received IByte2 stands for the 3rd byte received IByte3 stands for the 4th byte received, if CIM.DUMMY = 1, IByte3 will be 0. OByte0 stands for the 1st byte stored in memory OByte1 stands for the 2nd byte stored in memory OByte2 stands for the 3rd byte stored in memory OByte3 stands for the 4th byte stored in memory Then BS0 = 0: IByte0 is stored in OByte0 BS0 = 1: IByte1 is stored in OByte0	W
21:20	BS2		
19:18	BS1		
17:16	BS0		

		BS0 = 2: IByte2 is stored in OByte0 BS0 = 3: IByte3 is stored in OByte0 BS1 = 0: IByte0 is stored in OByte1 BS1 = 1: IByte1 is stored in OByte1 BS1 = 2: IByte2 is stored in OByte1 BS1 = 3: IByte3 is stored in OByte1 BS2 = 0: IByte0 is stored in OByte2 BS2 = 1: IByte1 is stored in OByte2 BS2 = 2: IByte2 is stored in OByte2 BS2 = 3: IByte3 is stored in OByte2 BS3 = 0: IByte0 is stored in OByte3 BS3 = 1: IByte1 is stored in OByte3 BS3 = 2: IByte2 is stored in OByte3 BS3 = 3: IByte3 is stored in OByte3 If CIMCFG.PACK is not 3'b100, BS0 should be 0, BS1 should be 1, BS2 should be 2 and BS3 should be 3.	
15	INV_DAT	Inverse every bit of input data. 0: not inverse; 1: inverse.	R W
14	VSP	VSYNC polarity selection. When VSYNC signal is input from pin. VSYNC, this bit specifies the VSYNC signal active level and leading edge. When VSYNC is retrieved from SAV&EAV, this bit is ignored. 0: VSYNC signal active high, VSYNC signal leading edge is rising edge 1: VSYNC signal active low, VSYNC signal leading edge is falling edge	R W
13	HSP	Specifies the HSYNC signal active level and leading edge. 0: HSYNC signal active high, HSYNC signal leading edge is rising edge 1: HSYNC signal active low, HSYNC signal leading edge is falling edge	R W
12	PCP	Specifies the PCLK working edge. 0: Data is sampled by PCLK rising edge 1: Data is sampled by PCLK falling edge	R W
11:10	BURST_ TYPE	DMA burst type. 00: INCR8 01: INCR16 10: INCR32 11: INCR64	R W
9	DUMMY	DUMMY zero function. When DUMMY is 1, CIM hardware adds one byte zero to every 3 input data bytes to form 32-bit data. 0: DUMMY zero function disabled 1: DUMMY zero function enabled	RW
8	E_VSYN	External / internal VSYNC selection. When DSM is ITU656Progressive	RW

<table>
<tr><td></td><td>C</td><td>Mode, VSYNC can be external (provided by sensor) or internal (retrieved from SAV&EAV). This bit only valid for ITU656Progressive Mode; In other DSM modes, this bit should always be 0.
0: Internal VSYNC mode, pin VSYNC is ignored
1: External VSYNC mode, VSYNC is provided by image sensor via pin VSYNC</td><td></td></tr>
<tr><td>7</td><td>LM</td><td>Line Mode for ITU656.
0: EAV is before SAV in each line
1: SAV is before EAV in each line</td><td>RW</td></tr>
<tr><td>6:4</td><td>PACK</td><td>Data packing mode, pack 8-bit input data into 32-bit data for FIFO.
<table>
<tr><th>PACK</th><th>Bypass Mode</th></tr>
<tr><td>3’b000</td><td>0x 11 22 33 44</td></tr>
<tr><td>3’b001</td><td>0x 22 33 44 11</td></tr>
<tr><td>3’b010</td><td>0x 33 44 11 22</td></tr>
<tr><td>3’b011</td><td>0x 44 11 22 33</td></tr>
<tr><td>3’b100</td><td>0x 44 33 22 11</td></tr>
<tr><td>3’b101</td><td>0x 33 22 11 44</td></tr>
<tr><td>3’b110</td><td>0x 22 11 44 33</td></tr>
<tr><td>3’b111</td><td>0x 11 44 33 22</td></tr>
</table>
In this table, 0x11, 0x22, 0x33 and 0x44 mean the received data from the sensor, 0x11 is received first and 0x44 is received last.
If the packed data does not meet wanted, it is recommended that
1) set PACK to 3’b100
2) use CIMCFG.BS0-3 to reorder the bytes.</td><td>6:4</td></tr>
<tr><td>3</td><td>FP</td><td>Field flag polarity selection. When ITU656 progressive stream is input, this bit specifies the field flag active level. When other modes are used, this bit is ignored.
0: Field flag active low
1: Field flag active high</td><td>RW</td></tr>
<tr><td>2</td><td>E_HSYNC</td><td>External / internal HSYNC selection. When DSM is ITU656Progressive Mode, HSYNC can be external (provided by sensor) or internal (retrieved from SAV&EAV). This bit only valid for ITU656Progressive Mode; In other DSM modes, this bit should always be 0.
0: Internal HSYNC mode, pin HSYNC is ignored
1: External HSYNC mode, HSYNC is provided by image sensor via pin HSYNC</td><td>RW</td></tr>
<tr><td>1:0</td><td>DSM</td><td>Data sample mode. Please refer to the table below.
<table>
<tr><th>DSM</th><th>Description</th></tr>
<tr><td>2’b00</td><td>ITU656Progressive Mode</td></tr>
<tr><td>2’b01</td><td>ITU656Interlace Mode</td></tr>
<tr><td>2’b10</td><td>Gated Clock Mode</td></tr>
<tr><td>2’b11</td><td>Reserved</td></tr>
</table></td><td>RW</td></tr>
</table>

5.2.2 CIM Control Register (CIMCR)

	CIMCR																													0x0004
Bit	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
	EEOF_LINE												FRC				Reserved	WINE	Reserved					FRM_ALIGN	DMA_SYNC	Reserved		STP_REQ	SW_RST	DMA_EN	RF_RST	ENA
RST	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

Bits	Name	Description	RW
31:20	EEOF_LINE	When EEOF_LINE lines data has been transferred of a frame, the EEOF flag will be set, and the EEOF interrupt will occur.	RW
19:16	FRC	CIM frame rate control. If FRC = n, CIM sampling one frame from every (n+1) frames from the sensor.	RW
15	Reserved	Read as zero.	R
14	WINE	To enable window-image. Used to indicate whether the registers CIMWSIZE and CIMWOFFSET work or not. 0: the value in CIMWSIZE and CIMWOFFSET will be ignored 1: the value in CIMWSIZE and CIMWOFFSET will be used	RW
13:9	Reserved	Read as zero.	R
8	FRM_ALIGN	The frame start address is forced to double-world aligned or not. 0: The frame start address is not forced to double-world aligned. 1: The frame start address is forced to double-world aligned.	RW
7	DMA_SYNC	The control bit to enable DMA synchronization. 0: The valid data input to CIM will be transferred by DMA to external memory 1: When a new descriptor-DMA transfer starts with writing CIMDA, a frame synchronization will be done, and the data in RXFIFO will be ignored	RW
6:5	Reserved	Read as zero.	R
4	STP_REQ	Stop request. This function is used when you want to disable or reset CIM controller which may be busy. When STP_REQ is sent to CIM controller, the controller will finish the current transfer and stop at idle, the corresponding status or interrupt can be seen. 0: Not request to stop 1: Request to stop	R
3	SW_RST	Software reset enable. 0: Don't care 1: Reset the CIM module.	RW

2	DMA_EN	Enable / disable the DMA function. 0: disable DMA; 1: enable DMA.	RW
1	RF_RST	RXFIFO software reset. Setting 1 to RXF_RST can reset RXFIFO immediately. Setting 0 to RXF_RST can stop resetting RXFIFO. After reset, RXFIFO is empty.	RW
0	ENA	Enable or disable the CIM module. 0: CIM is not enabled, or disable CIM immediately 1: CIM is enabled, or enabling CIM	RW

5.2.3 CIM Control Register 2 (CIMCR2)

	CIMCR2																													0x0050
Bit	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
	Reserved								FSC	ARIF	Reserved																OP		Reserved	OPE	Reserved	APM
RST	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

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23	FSC	Enable frame size check. 0: Disable frame size check. 1: Enable frame size check.	RW
22	ARIF	Enable auto recovery for incomplete frame. This field will effect only when FSC is 1. 0: Disable auto recovery. 1: Enable auto recovery.	RW
21:6	Reserved	Writing has no effect, read as zero.	R
5:4	OP	Optional Priority Configuration. Only used when OPE is set to 1.	RW

OP	CIM AHB Priority	Number of Data in FIFO
2'b00	0 1 2 3	n <= 8 8 < n <= 16 16 < n <= 32 32 < n
2'b01	0 1 2 3	n <= 16 16 < n <= 32 32 < n <= 64 64 < n

		2'b10 / 0 / n <= 32 1 / 32 < n <= 64 2 / 64 < n <= 96 3 / 96 < n 2'b11 / 0 / n <= 64 1 / 64 < n <= 96 2 / 96 < n <= 128 3 / 128 < n It is suggested to use 2'b10.	
3	Reserved	Writing has no effect, read as zero.	R
2	OPE	Optional Priority Mode Enable Control. Only used when APM is 1. 0: CIM calculates the priority according to the fifo status 1: CIM calculates the priority according to OPG which is configured by software	RW
1	Reserved	Writing has no effect, read as zero.	R
0	APM	Auto Priority Mode Enable Control. 0: Auto priority mode disable. CIM uses the priority set by arbiter 1: Auto priority mode enable. CIM can use the priority according the fifo status	RW

5.2.4 CIM Status Register (CIMST)

	CIMST																													0x0008
Bit	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
	Reserved																				DEEOF	DSTP	DEOF	DSOF	Reserved				FSE	RFOF	RFE	STP_ACK
RST	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	0

Bits	Name	Description	RW
31:12	Reserved	Writing has no effect, read as zero.	R
11	DEEOF	When set to 1, indicates the DMA has transferred CIMCTRL.EEOF_LINE lines data of a frame. Can generate an interrupt if CIMIMR.DEEOFM is 0. Write 0 to clear.	RW
10	DSTOP	When set to 1, indicates the DMA complete transferring data and stop the operation. Can generate an interrupt if CIMIMR.DSTPM is 0. Write 0 to clear.	RW

9	DEOF	When set to 1, indicates the DMA complete a transfer from RXFIFO to a frame buffer. Can generate an interrupt if CIMIMR. DEOFM is 0. Write 0 to clear.	RW
8	DSOF	When set to 1, Indicates the DMA start a transfer from RXFIFO to a frame buffer. Can generate an interrupt if CIMIMR. DSOFM is 0. Write 0 to clear.	RW
7:4	Reserved	Read as zeros.	R
3	FSE	Indicates that frame size check error. 0: No frame size check error occurs. 1: The received size of frame is not equal to the expected. Can generate an interrupt if CIMIMR.FSEM is 0. Write 0 to clear.	RW
2	RFOF	Indicates RXFIFO overflow. 0: RXFIFO is not overflow. 1: RXFIFO is overflow. Can generate an interrupt if CIMIMR.ROFM is 0. Write 0 to clear.	RW
1	RFE	Indicates RXFIFO is empty. 0: RXFIFO is not empty 1: RXFIFO is empty	R
0	STP_ACK	Indicate that the controller has entered into idle status when CIMCR.STP_REQ is 1. 0: CIM has not stopped. 1: CIM has stopped. Can generate an interrupt if CIMMR.STPM is 0. Write 0 to this bit to clear.	RW

5.2.5 CIM Interrupt Mask Register (CIMIMR)

	CIMIMR																												0x0058	
Bit	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
	Reserved																				DEEOFM	DSTPM	DEOFM	DSOFM	Reserved				FSEM	RFOFM	Reserved	STPM
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	0	1

Bits	Name	Description	RW
31:12	Reserved	Read as zeros.	R
11	DEEOFM	The control bit to mask DMA EEOF interrupt. 0: enable	RW

		1: mask	
10	DSTPM	The control bit to mask DMA STOP interrupt. 0: enable 1: mask	RW
9	DEOFM	The control bit to mask DMA EOF interrupt. 0: enable 1: mask	RW
8	DSOFM	The control bit to mask DMA SOF interrupt. 0: enable 1: mask	RW
7:4	Reserved	Read as zeros.	R
3	FSEM	The control bit to mask frame size check error interrupt. 0: enable 1: mask	RW
2	RFOFM	The control bit to mask RXFIFO overflow interrupt. 0: enable 1: mask	RW
1	Reserved	Read as zeros.	R
0	STPM	The control bit to mask STP_ACK interrupt. 0: enable 1: mask	RW

5.2.6 CIM Interrupt ID Register (CIMIID)

	CIMIID																														0x000C
Bit	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
	FID																															
RST	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

Bits	Name	Description	RW
31:0	FID	Interrupt frame ID. Contains a copy of the Frame ID register (CIMFID) from the descriptor currently being processed when a DMA_SOF or DMA_EOF interrupt is generated. CIMIID is written to only when CIMCMD.SOFINT or CIMCMD.EOFINT is high. As such, the register is considered to be sticky and will be overwritten only when the associated interrupt is cleared by writing the CIM state register.	R

5.2.7 CIM Descriptor Address (CIMDA)

	CIMIDA																														0x0020	
Bit	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
	NDA																															
RST	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

Bits	Name	Description	RW
31:0	NDA	Next descriptor physical address in external memory. DMAC gets the next descriptor according to it after finishing the current one. The target address Bits [3:0] must be zero to be aligned to 16-byte boundary.	RW

5.2.8 CIM Frame buffer Address Register (CIMFA)

	CIMIFA																														0x0024	
Bit	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
	FA																															
RST	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

Bits	Name	Description	RW
31:0	FA	Frame buffer virtual address in external memory when CIMCFG. SEP is 0. When starts CIM, DMA transfers data from RXFIFO to frame buffer. This address is increased by hardware automatically. Bits [6:0] must be zero to be aligned to 32-word boundary.	R

NOTE: CIMFA comes from DMA Descriptor, so here it is read-only.

5.2.9 CIM Frame ID Register (CIMFID)

	CIMFID																														0x0028	
Bit	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
	FID																															
RST	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

Bits	Name	Description	RW
31:0	FID	Frame ID. The particular use of this field is up to the software. This ID will be copied to the CIMIID register when an interrupt occurs.	R

NOTE: CIMFID comes from DMA Descriptor, so here it is read-only.

5.2.10 CIM DMA Command Register (CIMCMD)

	CIMFID																														0x002C
Bit	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
	SOFINT	EOFINT	EEOFINTE	STOP	Reserved												LEN															
RST	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

Bits	Name	Description	RW
31	SOFINTE	Interrupt enable for DMA starting a frame-buffer transfer. 1: DMA will set CIMSTATE.DMA_SOF when start of a frame-buffer transfer When one frame uses several buffers, it is suggested to set SOFINTE of first buffer only.	R
30	EOFINTE	Interrupt enable for DMA ending a frame-buffer transfer. 1: DMA will set CIMSTATE.DMA_EOF when CIMCMD.LEN is decreased to 0, which means end of a frame-buffer transfer When one frame uses several buffers, it is suggested to set EOFINTE of last buffer only.	R
29	EEOFINTE	Interrupt enable for DMA issuing an earlier eof interrupt.	R
28	STOP	DMA stop. When DMA complete transferring data, STOP bit decides whether DMA should loading next descriptor or not. 0: DMA start loading next descriptor 1: DMA stopped, and CIMSTATE.DMA_STOP bit is set 1 by hardware	R
27	OFAR	Auto recovery enable when there is RXFIFO overflow. 0: No auto recovery when overflow occurs, thus the software should do something 1: Auto recovery enable, the hardware will correct the overflow	R
26:24	Reserved	Writing has no effect, read as zero.	R
23:0	LEN	Length of transfer in words. Indicate the number of words to be transferred by DMA to a frame buffer. LEN = 0 is not valid. DMA transfers data according to LEN. Each time one or more word(s) been transferred, LEN is decreased automatically.	R

5.2.11 CIM Window Size (CIMWSIZE)

	CIMWSIZE																0x0030															
Bit	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
	Reserved			LPF													Reserved			PPL													
RST	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

Bits	Name	Description	RW
31:29	Reserved	Writing has no effect, read as zero.	R
28:16	LPF	Lines per frame for CIM output.	RW
15:13	Reserved	Writing has no effect, read as zero.	R
12:0	PPL	PPL must be multiples of 2. In fact, the number of CIM output data in word is equal to PPL/2.	RW

5.2.12 CIM Window Offset (CIMWOFFSET)

	CIMWOFFSET																0x0034															
Bit	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
	Reserved			V_OFFSET													Reserved			H_OFFSET													
RST	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

Bits	Name	Description	RW
31:29	Reserved	Writing has no effect, read as zero.	R
28:16	V_OFFSET	Vertical offset.	RW
15:13	Reserved	Writing has no effect, read as zero.	R
12:0	H_OFFSET	Horizontal offset. It should be an even number.	RW

5.2.13 CIM Frame Size Register (CIMFS)

This register will indicate how many pixels and bytes are included in a frame.

	CIMFS																0x0054															
Bit	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
	Reserved			FVS													BPP		Reserved	FHS												
RST	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

Bits	Name	Description	RW
31:29	Reserved	Writing has no effect, read as zero.	R
28:16	FVS	Vertical size of the frame N indicates the vertical includes (n+1) pixels.	RW
15:14	BPP	Number of bytes per pixel. 00: One pixel includes 1 byte. 01: One pixel includes 2 bytes. 10: One pixel includes 3 bytes. 11: One pixel includes 4 bytes.	RW
13	Reserved	Writing has no effect, read as zero.	R
12:0	FHS	Horizontal size of the frame. N indicates the horizontal includes (n+1) pixels.	RW

5.3 CIM Data Sample Modes

CIM controller supports several types of sample mode. The modes and the corresponding signals used are shown in the following diagram:

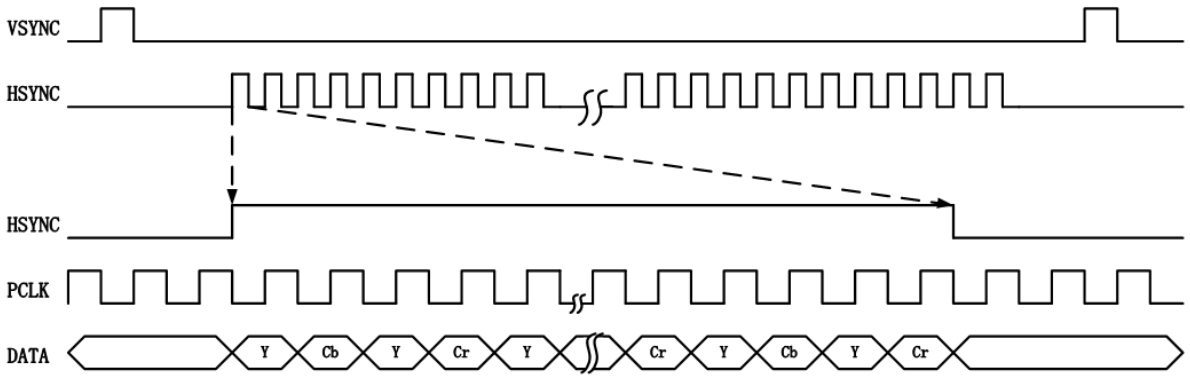
Table 5-3 The modes and the corresponding signals used

Mode \ Signals	VSYNC	HSYNC	PCLK	DATA
Gated Clock Mode	Y	Y	Y	Y
ITU656 Interlace Mode	N	N	Y	Y
ITU656 Progressive Mode	N	N	Y	Y

5.3.1 Gated Clock Mode

VSYNC, HSYNC, and PCLK signals are used in this mode.

A frame starts with VSYNC leading edge, then HSYNC goes active and holds the entire line. Data is sampled at the valid edge of PCLK when HSYNC is active; That means, HSYNC functions like “data enable” signal. Please refer to the figure below.



The VSYNC leading edge, HSYNC active level, PCLK valid edges are programmable.

5.3.2 ITU656 Interlace Mode

In this mode, PCLK and DATA signals are used, VSYNC, HSYNC signals are ignored.

CIM utilizes the SAV and EAV code within ITU656 data stream to get active video data.

The following diagrams and tables are quoted from ITU656 standard. For more information about ITU656, please refer to ITU656 standard.

For typical 625-line system and 525-line system, the field interval definitions are shown as flow:

		625	525
V-digital field blanking			
Field 1	Start (V = 1)	Line 624	Line 1
	Finish (V = 0)	Line 23	Line 20
Field 2	Start (V = 1)	Line 311	Line 264
	Finish (V = 0)	Line 336	Line 283
F-digital field identification			
Field 1	F = 0	Line 1	Line 4
Field 2	F = 1	Line 313	Line 266

Note:

F = 0, during field 0
F = 1, during field 1
V = 0, for blanking
V = 1, for video data
H = 0, SAV
H = 1, EAV

Timing reference code for 625/50 Video Systems:

LINE NUMBER	F	V	H (EAV)	H (SAV)	P0, P1, P2, P3
1-22	0 Field 1	1: blanking	1: in EAV, to indicate the end of active video	0: in SAV, to indicate the start of active video	Protection bits
23-310		0: video data			
311-312		1: blanking			
313-335	1 Field 2	1: blanking			
336-623		0: video data			
624-625		1: blanking			

Coding for protection codes:

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

5.3.3 ITU656 Progressive Mode

PCLK and DATA signals are used in this mode. HSYNC signal is ignored.

VSYNC is optional in this mode. When the start of frame information is retrieved from SAV and EAV, it is known as internal VSYNC mode. When VSYNC is provided by sensor directly, it is known as external VSYNC mode. CIM supports both internal and external VSYNC modes.

ITU656Progressive Mode is a kind of Non-Interlace Mode. The image data are encoded within only one field. Most sensors support ITU656Progressive Mode.

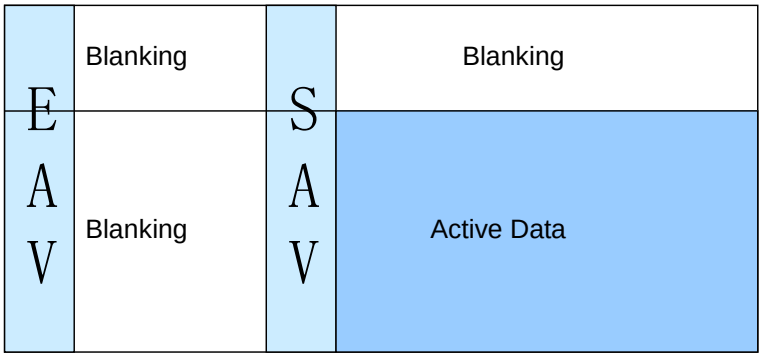


Figure 5-1 ITU656 Progressive Mode

5.4 DMA Descriptors

Used when output is packaged frame format.

A DMA descriptor is a 4-word block corresponding to the four DMA registers – CIMDA, CIMFA,

CIMFID, and CIMCMD, aligned on 4-word (16-byte) boundary, in external memory:

word [0] contains the physical address for next CIMDA
word [1] contains the value for CIMFID
word [2] contains the physical address for CIMFA
word [3] contains the value for CIMCMD

Software must write the physical address of the first descriptor to CIMDA before enabling the CIM. Once the CIM is enabled, the first descriptor is read, and all 4 registers are written by the DMAC. The next DMA descriptor pointed to by CIMDA is loaded into the registers after all data for the current descriptor has been transferred.

5.5 Software Operation

5.5.1 Enable CIM with DMA

Configure register CIMCFG.
Prepare frame buffer and descriptors.
Configure register CIMDA.
Clear state register: write 0 to register CIMSTATE.
Reset RXFIFO: configure register CIMCTRL with DMA_EN=1, RXF_RST=1, ENA=0.
Stop resetting RXFIFO: configure register CIMCTRL with DMA_EN=1, RXF_RST=0, ENA=0.
Enable CIM: configure register CIMCTRL with DMA_EN=1, RXF_RST=0, ENA=1.

5.5.2 Operations for RXFIFO Overflow

When using software to deal with RXFIFO overflow, the flows steps is suggested:

Step 1. Request CIM controller enter into idle state. Set CIMCR.STP_REQ to 1.
Step 2. Wait CIM controller enter into idle state. Waiting for CIMST.STP_ACK to 1.
Step 3. Reset CIM controller. Using CIMCR.SW_RST.
Step 4. Initialize CIM controller and start the operations needed.

When using hardware to deal with RXFIFO overflow,the flows configuration should be used:

CIMCMD.OFAR should be set to 1 when preparing the DMA descriptor(s).

5.5.3 Operations for Frame Size Error

When using software to deal with frame size error, the flows steps is suggested:

Step 1. Request CIM controller enter into idle state. Set CIMCR.STP_REQ to 1.
Step 2. Wait CIM controller enter into idle state. Waiting for CIMST.STP_ACK to 1.

Step 3. Reset CIM controller. Using CIMCR.SW_RST.

Step 4. Initialize CIM controller and start the operations needed.

When using hardware to deal with frame size error, the flows configuration should be used:

CIMCMD.OFAR should be set to 1 when preparing the DMA descriptor(s).

6 Audio Interface Controller

6.1 Overview

This chapter describes the AIC (I2S and SPDIF Controller) included in this processor.

The AIC supports the I2S or IIS (for inter-IC sound), a protocol defined by Philips Semiconductor. Both normal I2S and the MSB-justified I2S formats are supported by AIC.

AIC consists of buffers, status registers, control registers, serializers, and counters for transferring digitized audio between the processor's system memory, an external I2S CODEC. AIC can record digitized audio by storing the samples in system memory. For playback of digitized audio or production of synthesized audio, the AIC retrieves digitized audio samples from system memory and sends them to a CODEC through the serial connection with I2S formats. The external digital-to-analog converter in the CODEC then converts the audio samples into an analog audio waveform. The audio sample data can be stored to and retrieved from system memory either by the DMA controller or by programmed I/O.

Where both normal I2S and MSB-justified-I2S work with a variety of clock rates, which can be obtained either by dividing the PLL clock by two programmable dividers or from an external clock source.

SPDIF (Sony/Philips Digital Interconnect Format) is a digital audio interface. The transmission medium can be either electrical or optical. Supports IEC60958 two-channel PCM audio and IEC61937 multi-channel compressed audio (Dolby Digital, DTS, etc.).

This chapter describes the programming model for the AIC.

6.1.1 Block Diagram

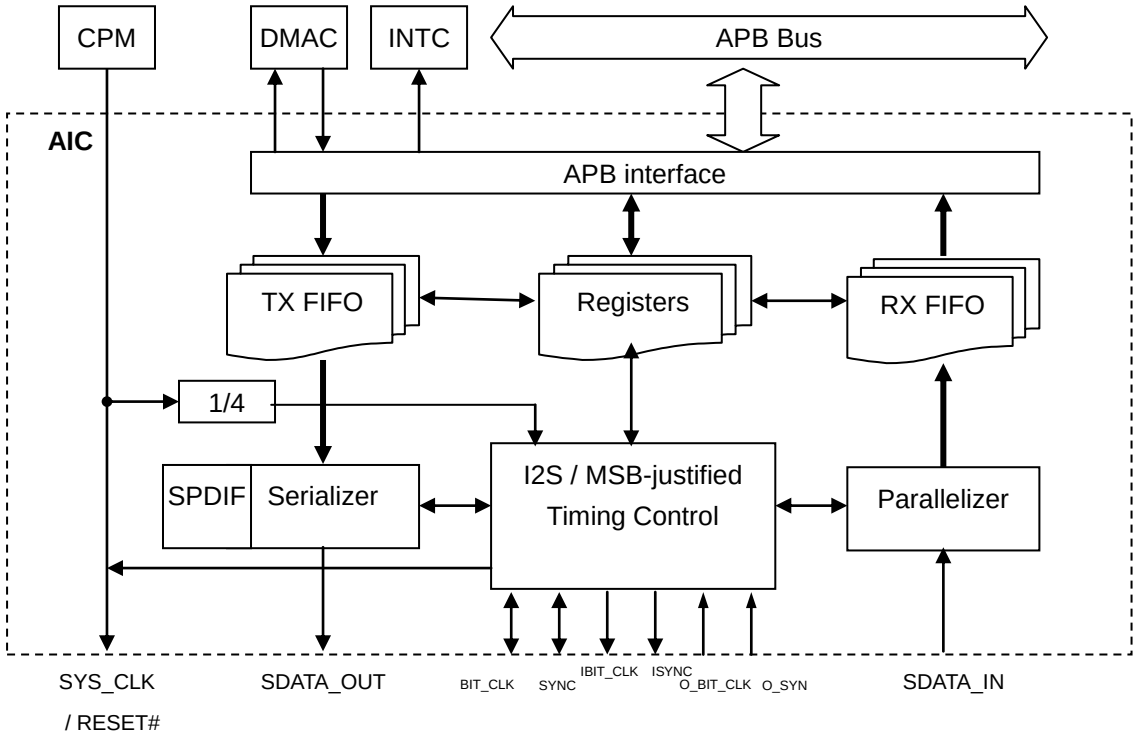


Figure 6-1 AIC Block Diagram

6.1.2 Features

AIC support following I2S/SPDIF features:

I2S features

- 8, 16, 18, 20 and 24 bit audio sample data sizes supported, 16 bits packed sample data is supported
- DMA transfer mode supported
- Stop serial clock supported
- Programmable Interrupt function supported
- Support mono PCM data to stereo PCM data expansion on audio play back
- Support endian switch on 16-bits normal audio samples play back
- Internal programmable or external serial clock and optional system clock supported for I2S or MSB-Justified format
- Two FIFOs for transmit and receive respectively

SPDIF features

- 8, 16, 18, 20 and 24 bit audio sample data sizes supported
- DMA transfer mode supported
- Stop serial clock supported
- Programmable Interrupt function supported

- Support IEC60958 two-channel PCM audio
- Support IEC61937 multi-channel compressed audio
- Support consumer mode and only support transmitter mode
- Profession mode is not supported
- The User data bit is ‘0’as it is not supported in the chip
- Support sampling frequency from 32kHz to 192kHz

6.1.3 Interface Diagram

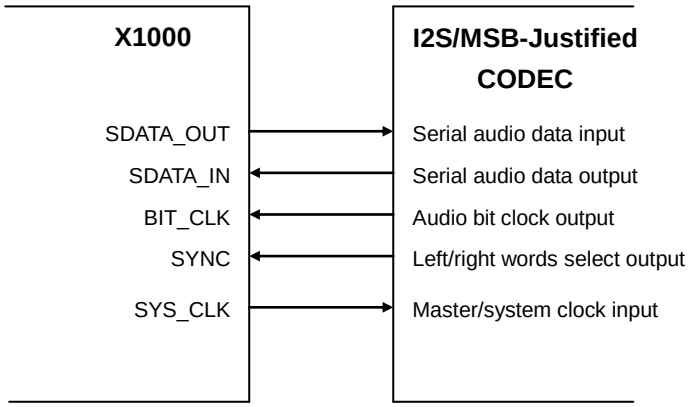


Figure 6-2 Interface to an External Master Mode I2S/MSB-Justified CODEC Diagram (Share Clock Mode)

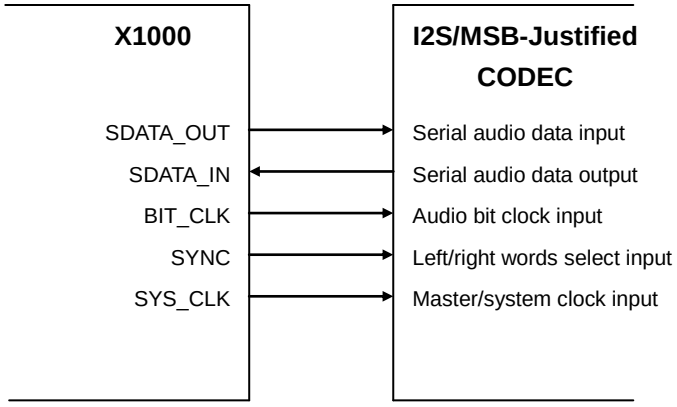


Figure 6-3 Interface to an External Slave Mode I2S/MSB-Justified CODEC Diagram (Share Clock Mode)

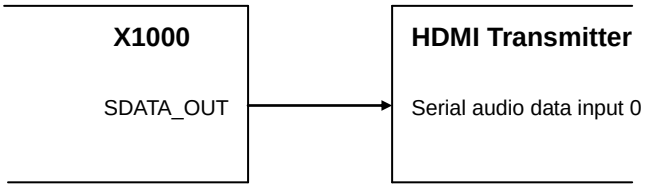


Figure 6-4 Interface to a HDMI Transmitter via SPDIF Diagram

6.1.4 Signal Descriptions

There are all 7 pins used to connect between AIC and an external audio CODEC device.

Table 6-1 AIC Pins Description

Function Name	PIN Name	I/O	Description
RESET# **SYS_CLK**	SCLK_RSTN	O	SYS_CLK: I2S/MSB-Justified formats, supply system clock to CODEC.
SDATA_IN	SDATI	I	Serial audio input data from CODEC.
BIT_CLK	BCLK	I/O	Sample rate dependent bit-rate clock input/output for I2S/MSB-Jistified.
BIT_CLK_ADC	ADCBCLK		Similarly as BIT_CLK, can only be use for ADC channel.
SYNC	LRCLK	I/O	Indicates the left- or right-channel for I2S/MSB-Justified format.
SYNC_ADC	ADCLRCLK		Similarly as SYNC, can only be use for ADC channel.
SDATA_OUT	SDATO	O	Serial audio output data to CODEC / I2S line 0 / SPDIF output.

6.1.4.1 SYS_CLK Pin

SYS_CLK outputs the system clock to CODEC. This pin is useful only in I2S/MSB-justified format. It generates a frequency between approximately 2.048 MHz and 24.576 MHz by dividing down the PLL clock with a programmable divisor. This frequency can be 256, 384, 512 and etc. times of the audio sampling frequency. Or it can be set to a wanted frequency. If AIC is disabled, it retains the high.

6.1.4.2 BIT_CLK Pin

BIT_CLK is the serial data bit rate clock, at which I2S data moves between the CODEC and the processor. One bit of the serial data is transmitted or received each BIT_CLK period. In I2S and MSB-justified format it inputs from the CODEC in slave mode and outputs to CODEC in master mode. In the master mode, the clock is generated internally that is 64 times the sampling frequency.

Table 6-5 ists the available sampling frequencies based on an internal clock source. If AIC is disabled, AICFR.AUSEL and AICFR.BCKD determine the direction. And it retains the low if it is output and the state is undefined if it is input.

The IBIT_CLK is for the SDATA_IN signal on division CLOCK function.

6.1.4.3 SYNC Pin

In I2S/MSB-Justified formats, SYNC is used to indicate left- or right-channel sample data and toggled in sample rate frequency. It outputs to CODEC in master mode and inputs from CODEC in slave mode. If AIC is disabled, AICFR.AUSEL and AICFR.BCKD determine the direction. And it retains the low if it is output and the state is undefined if it is input.

The ISYNC is for the SDATA_IN signal on division CLOCK function.

6.1.4.4 SDATA_OUT Pin

SDATA_OUT is AIC output data pin, which outputs I2S serial audio data, SPDIF serial data to an external audio CODEC device.
If in multi channels mode, it outputs the first two channels serial data.
If AIC is disabled, it retains the low.

6.1.4.5 SDATA_IN Pin

SDATA_IN is AIC inputs data pin, which inputs serial audio data from an external audio CODEC device. If AIC is disabled, its state is undefined.

6.2 Register Descriptions

AIC software interface includes 13 registers and 1 FIFO data port. They are mapped in IO memory address space so that program can access them to control the operation of AIC and the outside CODEC.

Table 6-2 AIC Registers Description

Name	Description	RW	Reset value	Address	Size
AICFR	AIC Configuration Register	RW	0x07100000	0x10020000	32
AICCR	AIC Common Control Register	RW	0x01240000	0x10020004	32
I2SCR	AIC I2S/MSB-justified Control Register	RW	0x00000000	0x10020010	32
AICSR	AIC FIFO Status Register	RW	0x00000008	0x10020014	32
I2SDIV	AIC I2S/MSB-justified Clock Divider Register	RW	0x00000003	0x10020030	32
AICDR	AIC FIFO Data Port Register	RW	0x????????	0x10020034	32

SPENA	SPDIF Enable Register	RW	0x00	0x10020080	8
SPCTRL	SPDIF Control Register	RW	0x0003	0x10020084	16
SPSTATE	SPDIF Status Register	RW	0x0000	0x10020088	16
SPCFG1	SPDIF Configure 1 Register	RW	0x00000000	0x1002008C	32
SPCFG2	SPDIF Configure 2 Register	RW	0x00000000	0x10020090	32
SPFIFO	SPDIF FIFO Register	W	0x????????	0x10020094	32

1 AICFR is used to control FIFO threshold, I2S/MSB-justified selection, AIC reset, master/slave selection
2 AICCR is used to control DMA mode, FIFO flush, interrupt enable, internal loop-back, play back and recording enable. It also controls sample size and signed/unsigned data transfer.
3 I2SCR is used to control BIT_CLK stop, audio sample size, I2S or MSB-justified selection in I2S/MSB-justified.
4 AICSR is used to reflect FIFOs status.
5 I2SSR is used to reflect AIC status in I2S/MSB-justified.
6 I2SDIV is used to set clock divider for BIT_CLK generating in I2S/MSB-justified format.
7 AICDR is act as data input/output port to/from transmit/receive FIFO when write/read..

6.2.1 AIC Configuration Register (AICFR)

AICFR contains bits to control FIFO threshold, I2S/MSB-justified selection, AIC reset, master/slave selection, and AIC enable.

	AICFR																															0x10020000
Bit	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
	Reserved				RFTH				Reserved			TFTH					Reserved			MSB	Reserved	IBCKD	ISYNCD	DMODE	CDC_SLAVE	LSMP	ICDC	AUSEL	RST	BCKD	SYNCD	ENB
RST	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:24	RFTH	Receive FIFO threshold for interrupt or DMA request. The RFTH valid value is 0 ~ 15. This value represents a threshold value of (RFTH + 1) * 2. When the sample number in receive FIFO, indicated by AICSR.RFL, is great than or equal to the threshold value, AICSR.RFS is set. Larger RFTH value provides lower DMA/interrupt request frequency but have more risk to involve receive FIFO overflow. The optimum value is system dependent.	RW
23:21	Reserved	Writing has no effect, read as zero.	R
20:16	TFTH	Transmit FIFO threshold for interrupt or DMA request. The TFTH valid value 0 ~ 31.	RW

<table>
<tr><td></td><td></td><td>This value represents a threshold value of TFTH * 2. When the sample number in transmit FIFO, indicated by AICSR.TFL, is less than or equal to the threshold value, AICSR.TFS is set. Smaller TFTH value provides lower DMA/interrupt request frequency but have more risk to involve transmit FIFO underflow. The optimum value is system dependent.</td><td></td></tr>
<tr><td>15:13</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>12</td><td>MSB</td><td>The big-endian format enable. When this bit is '1', the replay data will use the big-endian format. The sign bit at the bit 31. Else the replay data is little-endian format.</td><td>RW</td></tr>
<tr><td>11</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>10</td><td>IBCKD</td><td>Keep this value to 0 in normal use.</td><td>RW</td></tr>
<tr><td>9</td><td>ISYNCD</td><td>Keep this value to 0 in normal use.</td><td>RW</td></tr>
<tr><td>8</td><td>DMODE</td><td>Keep this value to 0 in normal use.</td><td>RW</td></tr>
<tr><td>7</td><td>CDC_
SLAVE</td><td>Keep this value to 0 in normal use.</td><td>R</td></tr>
<tr><td>6</td><td>LSMP</td><td>Select between play last sample or play ZERO sample in TX FIFO underflow. ZERO sample means sample value is zero. This bit is better be changed while audio replay is stopped.
<table>
<tr><th>LSMP</th><th>CODEC used</th></tr>
<tr><td>0</td><td>Play ZERO sample when TX FIFO underflow.</td></tr>
<tr><td>1</td><td>Play last sample when TX FIFO underflow.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>5</td><td>ICDC</td><td><table>
<tr><th>ICDC</th><th>CODEC used</th></tr>
<tr><td>0</td><td>External CODEC.</td></tr>
<tr><td>1</td><td>Reserved</td></tr>
</table></td><td>RW</td></tr>
<tr><td>4</td><td>AUSEL</td><td>Audio Unit Select. Change this bit in case of BIT_CLK is stopped (I2SCR.STPBK = 1).
<table>
<tr><th>AUSEL</th><th>Selected</th></tr>
<tr><td>0</td><td>Reserved</td></tr>
<tr><td>1</td><td>Select I2S/MSB-justified format.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>3</td><td>RST</td><td>Reset AIC. Write 1 to this bit reset AIC registers and FIFO except AICFR and I2SDIV register. Writing 0 to this bit has no effect and this bit is always reading 0.</td><td>W</td></tr>
<tr><td>2</td><td>BCKD</td><td>BIT_CLK Direction. This bit specifies input/output direction of BIT_CLK. It is only valid in I2S/MSB-justified format. Change this bit in case of BIT_CLK is stopped (I2SCR.STPBK = 1).
<table>
<tr><th>BCKD</th><th>BIT_CLK Direction</th></tr>
<tr><td>0</td><td>BIT_CLK is input from an external source.</td></tr>
<tr><td>1</td><td>BIT_CLK is generated internally and driven out to the CODEC.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>1</td><td>SYNCD</td><td>SYNC Direction. This bit specifies input/output direction of SYNC in I2S/MSB-justified format. Change this bit in case of BIT_CLK is stopped</td><td>RW</td></tr>
</table>

		(I2SCR.STPBK = 1). **SYNCD** / **SYNC Direction** 0 – SYNC is input from an external source. 1 – SYNC is generated internally and driven out to the CODEC.	
0	ENB	Enable AIC function. This bit is used to enable or disable the AIC function. **ENB** / **Description** 0 – **Disable AIC Controller.** 1 – **Enable AIC Controller.**	RW

The BCKD bit (bit 2) and SYNCD bit (bit 1) configure the mode of I2S/MSB-justified interface. This is compliant with I2S specification.

BCKD	SYNCD	Description
0 (input)	**0 (input)**	**AIC roles the slave of I2S/MSB-justified interface.**
	1 (output)	**AIC roles the master with external serial clock source of I2S/MSB-justified interface.**
1 (output)	**0 (input)**	**Reserved.**
	1 (output)	**AIC roles the master of I2S/MSB-justified interface.**

6.2.2 AIC Common Control Register (AICCR)

AICCR contains bits to control DMA mode, FIFO flush, interrupt enable, internal loop-back, play back and recording enable. It also controls sample size and signed/unsigned data transfer.

AICCR **0x10020004**

Bit	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
	Reserved		Reserved	PACK16	Reserved	CHANNEL			Reserved		OSS			ISS			RDMS	TDMS	Reserved		M2S	ENDSW	ASVTSU	TFLUSH	RFLUSH	EROR	ETUR	ERFS	ETFS	ENLBF	ERPL	EREC
RST	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:30	Reserved	Writing has no effect, read as zero.	R
29	Reserved	Keep this value to 0 in normal use.	R
28	PACK16	Output Sample data 16bit packed mode select. This bit reflects that one word contains two sample data or only one sample data with LSB align. The packed mode is only support 16bit sample size. **PACK16** / **Sample Size** 0 – Unpacked data mode. One word only contains one 16bit sample	RW

<table>
<tr><td></td><td></td><td></td><td><table><tr><td></td><td>data aligned LSB.</td></tr><tr><td>1</td><td>Packed data mode.
One word contains two 16 bit sample data.</td></tr></table></td><td></td></tr>
<tr><td>27</td><td>Reserved</td><td colspan="2">Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>26:24</td><td>CHANNEL</td><td colspan="2">Output Channel Number Select. These bits reflect output data channels from AIC to device. The data supported are: 1(mono), 2(stereo), 4, 6 and 8 channels. The sample data is LSB-justified in memory/register.<table><tr><th>CHANNEL</th><th>Sample Size</th></tr><tr><td>0x0</td><td>1 channel, mono mode.</td></tr><tr><td>0x1</td><td>2 channels, stereo mode.</td></tr><tr><td>0x2 ~ 0x7</td><td>Reserved.</td></tr></table></td><td>RW</td></tr>
<tr><td>23:22</td><td>Reserved</td><td colspan="2">Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>21:19</td><td>OSS</td><td colspan="2">Output Sample Size. These bits reflect output sample data size from memory or register. The data sizes supported are: 8, 16, 18, 20 and 24 bits. The sample data is LSB-justified in memory/register.<table><tr><th>OSS</th><th>Sample Size</th></tr><tr><td>0x0</td><td>8 bit.</td></tr><tr><td>0x1</td><td>16 bit.</td></tr><tr><td>0x2</td><td>18 bit.</td></tr><tr><td>0x3</td><td>20 bit.</td></tr><tr><td>0x4</td><td>24 bit.</td></tr><tr><td>0x5~0x7</td><td>Reserved.</td></tr></table></td><td>RW</td></tr>
<tr><td>18:16</td><td>ISS</td><td colspan="2">Input Sample Size. These bits reflect input sample data size to memory or register. The data sizes supported are: 8, 16, 18, 20 and 24 bits. The sample data is LSB-justified in memory/register.<table><tr><th>ISS</th><th>Sample Size</th></tr><tr><td>0x0</td><td>8 bit.</td></tr><tr><td>0x1</td><td>16 bit.</td></tr><tr><td>0x2</td><td>18 bit.</td></tr><tr><td>0x3</td><td>20 bit.</td></tr><tr><td>0x4</td><td>24 bit.</td></tr><tr><td>0x5~0x7</td><td>Reserved.</td></tr></table></td><td>RW</td></tr>
<tr><td>15</td><td>RDMS</td><td colspan="2">Receive DMA enable. This bit is used to enable or disable the DMA during receiving audio data.<table><tr><th>RDMS</th><th>Receive DMA</th></tr><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td><td>RW</td></tr>
<tr><td>14</td><td>TDMS</td><td colspan="2">Transmit DMA enable. This bit is used to enable or disable the DMA during transmit audio data.<table><tr><th>TDMS</th><th>Transmit DMA</th></tr><tr><td>0</td><td>Disabled.</td></tr><tr><td>1</td><td>Enabled.</td></tr></table></td><td>RW</td></tr>
</table>

<table>
<tr><td>13:12</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>11</td><td>M2S</td><td>Mono To Stereo. This bit control whether to do mono to stereo sample expansion in play back. When this bit is set, every outgoing sample data in the steam plays in both left and right channels. This bit should only be set in 2 channels configuration. It takes effective immediately when the bit is changed. Change this before replay started.
<table>
<tr><th>M2S</th><th>Description</th></tr>
<tr><td>0</td><td>No mono to stereo expansion.</td></tr>
<tr><td>1</td><td>Do mono to stereo expansion.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>10</td><td>ENDSW</td><td>Endian Switch. This bit control endian change on outgoing 16-bits size audio sample by swapping high and low bytes in the sample data.
<table>
<tr><th>ENDSW</th><th>Description</th></tr>
<tr><td>0</td><td>No change on outgoing sample data.</td></tr>
<tr><td>1</td><td>Swap high and low byte for outgoing 16-bits size sample data.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>9</td><td>ASVTSU</td><td>Audio Sample Value Transfer between Signed and Unsigned data format. This bit is used to control the signed ←→unsigned data transfer. If it is 1, the incoming and outgoing audio sample data will be transferred by toggle its most significant bit.
<table>
<tr><th>ASVTSU</th><th>Description</th></tr>
<tr><td>0</td><td>No audio sample value signed ←→unsigned transfer.</td></tr>
<tr><td>1</td><td>Do audio sample value signed ←→unsigned transfer.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>8</td><td>TFLUSH</td><td>Transmit FIFO Flush. Write 1 to this bit flush transmit FIFOs to empty. Writing 0 to this bit has no effect and this bit is always reading 0.</td><td>W</td></tr>
<tr><td>7</td><td>RFLUSH</td><td>Receive FIFO Flush. Write 1 to this bit flush receive FIFOs to empty. Writing 0 to this bit has no effect and this bit is always reading 0.</td><td>W</td></tr>
<tr><td>6</td><td>EROR</td><td>Enable ROR Interrupt. This bit is used to control the ROR interrupt enable or disable.
<table>
<tr><th>EROR</th><th>ROR Interrupt</th></tr>
<tr><td>0</td><td>Disabled.</td></tr>
<tr><td>1</td><td>Enabled.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>5</td><td>ETUR</td><td>Enable TUR Interrupt. This bit is used to control the TUR interrupt enable or disable.
<table>
<tr><th>ETUR</th><th>TUR Interrupt</th></tr>
<tr><td>0</td><td>Disabled.</td></tr>
<tr><td>1</td><td>Enabled.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>4</td><td>ERFS</td><td>Enable RFS Interrupt. This bit is used to control the RFS interrupt enable</td><td>RW</td></tr>
</table>

		or disable. **ERFS** / **RFS Interrupt** 0 / **Disabled.** 1 / **Enabled.**	
3	ETFS	Enable TFS Interrupt. This bit is used to control the TFS interrupt enable or disable. **ETFS** / **TFS Interrupt** 0 / **Disabled.** 1 / **Enabled.**	RW
2	ENLBF	Enable AIC Loop Back Function. This bit is used to enable or disable the internal loop back function of AIC, which is used for test only. When the AIC loop back function is enabled, normal audio replay/record functions are disabled. **ENLBF** / **Description** 0 / AIC Loop Back Function is Disabled. 1 / AIC Loop Back Function is Enabled.	RW
1	ERPL	Enable Playing Back function. This bit is used to disable or enable the audio sample data transmitting. **ERPL** / **Description** 0 / **AIC Playing Back Function is Disabled.** 1 / **AIC Playing Back Function is Enabled.**	RW
0	EREC	Enable Recording Function. This bit is used to disable or enable the audio sample data receiving. **EREC** / **Description** 0 / **AIC Recording Function is Disabled.** 1 / **AIC Recording Function is Enabled.**	RW

6.2.3 AIC I2S/MSB-justified Control Register (I2SCR)

I2SCR contains bits to control BIT_CLK stop, audio sample size, I2S or MSB-justified selection in I2S/MSB-justified. It is valid only in I2S/MSB-justified format.

I2SCR **0x10020010**

Bit	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
	Reserved														RFIRST	SWLH	Reserved		ISTPBK	STPBK	Reserved							ESCLK	Reserved			AMSL
RST	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

Bits	Name	Description	RW
31:18	Reserved	Writing has no effect, read as zero.	R

<table>
<tr><td>17</td><td>RFIRST</td><td>Send R channel first in stereo mode. This bit should only be set in 2 channels configuration. The frame is LR like or RL like. It takes effective immediately when the bit is changed.
Change this before replay started.
<table><tr><th>RFIRST</th><th>Description</th></tr><tr><td>0</td><td>Send L channel first. (LR)</td></tr><tr><td>1</td><td>Send R channel first. (RL)</td></tr></table></td><td>RW</td></tr>
<tr><td>16</td><td>SWLH</td><td>Switch LR channel in 16bit-packed stereo mode.
This bit control whether the low address data is L or R. This bit should only be set in 2 channels configuration and 16bit-packed mode. That means it can only valid with packed mode (PACK16 =1) and 2 channels (CHANNEL=0x1).
It takes effective immediately when the bit is changed. Change this before replay started.
<table><tr><th>SWLH</th><th>Description</th></tr><tr><td>0</td><td>16 bit LSB and 16bit MSB is not switched.</td></tr><tr><td>1</td><td>16 bit LSB and 16bit MSB is switched.</td></tr></table></td><td>RW</td></tr>
<tr><td>15:12</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>13</td><td>ISTPBK</td><td>Keep this value to 0 in normal use.</td><td>RW</td></tr>
<tr><td>12</td><td>STPBK</td><td>Stop BIT_CLK. It is used to stop the BIT_CLK in I2S/MSB-justified format.
<table><tr><th>STPBK</th><th>Description</th></tr><tr><td>0</td><td>BIT_CLK is not stopped.</td></tr><tr><td>1</td><td>BIT_CLK is stopped.</td></tr></table>
Please set this bit to 1 to stop BIT_CLK when change AICFR.AUSEL and AICFR.BCKD.</td><td>RW</td></tr>
<tr><td>11:5</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>4</td><td>ESCLK</td><td>Enable SYSCLK output. When this bit is 1, the SYSCLK outputs to chip outside is enabled. Else, the clock is disabled.</td><td>RW</td></tr>
<tr><td>3:1</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>0</td><td>AMSL</td><td>Specify Alternate Mode (I2S or MSB-Justified) Operation.
<table><tr><th>AMSL</th><th>Description</th></tr><tr><td>0</td><td>Select I2S Operation Mode.</td></tr><tr><td>1</td><td>Select MSB-Justified Operation Mode.</td></tr></table></td><td>RW</td></tr>
</table>

6.2.4 AIC Controller FIFO Status Register (AICSR)

AICSR contains bits to reflect FIFOs status. Most of the bits are read-only except two, which can be written a 0.

	AICSR																													0x10020014
Bit	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
	Reserved		RFL						Reserved								TFL								Reserved	ROR	TUR	RFS	TFS	Reserved		
RST	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

Bits	Name	Description	RW
31:30	Reserved	Writing has no effect, read as zero.	R
29:24	RFL	Receive FIFO Level. The bits indicate the amount of valid PCM data in Receive FIFO. **RFL Value** / **Description** 0x00 ~ 0x20 — RFL valid PCM data in receive FIFO. 0x21 ~ 0x3F — Reserved.	R
23:14	Reserved	Writing has no effect, read as zero.	R
13:8	TFL	Transmit FIFO Level. The bits indicate the amount of valid PCM data in Transmit FIFO. **TFL Value** / **Description** 0x00 ~ 0x3F — TFL valid PCM data in transmit FIFO.	R
7	Reserved	Writing has no effect, read as zero.	R
6	ROR	Receive FIFO Over Run. This bit indicates that receive FIFO has or has not experienced an overrun. **ROR** / **Description** 0 — When read, indicates over-run has not been found. 0 — When write, clear itself. 1 — When read, indicates data has even been written to full receive FIFO. 1 — When write, not effects.	RW
5	TUR	Transmit FIFO Under Run. This bit indicates that transmit FIFO has or has not experienced an under-run. **TUR** / **Description** 0 — When read, indicates under-run has not been found. 0 — When write, clear itself. 1 — When read, indicates data has even been read from empty transmit FIFO. 1 — When write, not effects.	RW
4	RFS	Receive FIFO Service Request. This bit indicates that receive FIFO level is or not below receive FIFO threshold, which is controlled by AICFR.RFTH. When RFS is 1, it may trigger interrupt or DMA request depends on the interrupt enable and DMA setting.	R

			RFS	**Description**
			0	Receive FIFO level below RFL threshold.
			1	Receive FIFO level at or above RFL threshold.
3	TFS	Transmit FIFO Service Request. This bit indicates that transmit FIFO level is below Transmit FIFO threshold, which is controlled by AICFR.TFTH. When TFS is 1, it may trigger interrupt or DMA request depends on the interrupt enable and DMA setting. **TFS** / **Description** 0 / Transmit FIFO level exceeds TFL threshold. 1 / Transmit FIFO level at or below TFL threshold.		R
2:0	Reserved	Writing has no effect, read as zero.		R

6.2.5 AIC I2S/MSB-justified Status Register (I2SSR)

I2SSR is used to reflect AIC status in I2S/MSB-justified. It is a read-only register.

I2SSR **0x1002001c**

Bit	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
	Reserved																										CHBSY	TBSY	RBSY	BSY	Reserved	
RST	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

Bits	Name	Description	RW
31:3	Reserved	Writing has no effect, read as zero.	R
5	CHBSY	AIC Transmitter busy in I2S/MSB-justified format.(Multi-channel status) **CHBSY** / **Description** 0 / AIC Transmitter part is idle or disabled. 1 / AIC Transmitter part currently is transmitting or receiving a frame.	R
4	TBSY	AIC Transmitter busy in I2S/MSB-justified format. **TBSY** / **Description** 0 / AIC Transmitter part is idle or disabled. 1 / AIC Transmitter part currently is transmitting or receiving a frame.	R
3	RBSY	AIC Receiver busy in I2S/MSB-justified format. **RBSY** / **Description** 0 / AIC Receiver part is idle or disabled. 1 / AIC Receiver part currently is transmitting or receiving a frame.	R
2	BSY	AIC busy in I2S/MSB-justified format. **BSY** / **Description**	R

		0	AIC controller is idle or disabled.
		1	AIC controller currently is transmitting or receiving a frame.
1:0	Reserved	Writing has no effect, read as zero.	R

6.2.6 AIC I2S/MSB-justified Clock Divider Register (I2SDIV)

I2SDIV is used to set clock divider to generated BIT_CLK from SYS_CLK in I2S/MSB-justified format.

	I2SDIV																															0x10020030
Bit	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
	Reserved																DV															
RST	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

Bits	Name	Description	RW
31:9	Reserved	Writing has no effect, read as zero.	R
8:0	DV	Audio BIT_CLK clock divider value minus 1. I2SDIV.DV is used to control the generating of BIT_CLK from dividing SYS_CLK. The dividing value can be even or odd and I2SDIV.IDV should be set to the dividing value minus 1. BIT_CLK frequency is fixed to 64 f_S in AIC, where f_S is the audio sample frequency. I2SDIV.DV depends on SYS_CLK frequency f_{SYS_CLK}, which is selected according to external CODEC's requirement and internal PLL frequency. Please reference to 1.4.10 Serial Audio Clocks and Sampling Frequencies for further description.	RW

6.2.7 AIC FIFO Data Port Register (AICDR)

AICDR is act as data input port to transmit FIFO when write and data output port from receive FIFO when read, one audio sample every time. The FIFO width is 24 bits. Audio sample with size N that is less than 24 is located in LSB N-bits. The sample size is specified by I2SCR.WL in I2S/MSB-justified. In I2S/MSB-justified, the left channel sample is prior to the right channel sample.

Care should be taken to monitor the status register to insure that there is room for data in the FIFO when executing a program read or write transaction. This is taken care automatically in DMA.

	AICDR																															0x10020034
Bit	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
	Reserved								DATA																							
RST	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:0	DATA	FIFO port. When write to it, data is push to the transmit FIFO. When read from it, data is pop from the receiving FIFO.	RW

6.2.8 SPDIF Enable Register (SPENA)

The register SPENA is used to trigger SPDIF transmitter to work.

	SPENA																								0x10020080							
Bit																									7	6	5	4	3	2	1	0
	Reserved																															SPEN
RST																									0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:1	Reserved	Writing has no effect, read as zero.	R
0	SPEN	Enable / disable the SPDIF transmitter. 0: SPDIF transmitter is disabled 1: SPDIF transmitter is enabled	RW

6.2.9 SPDIF Control Register (SPCTRL)

The register SPCTRL is used to control SPDIF to work.

	SPCTRL																0x10020084															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	DMA_EN	D_TYPE	SIGN_N	INVALID	SFT_RST	SPDIF_I2S	Reserved								M_TRIG	M_FFUR
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	Name	Description	RW
15	DMA_EN	DMA transmitter enable bit. 0: DMA transmitter disable 1: DMA transmitter enable	RW
14	D_TYPE	If the bit number of data is less than16, the data in memory is as follows: 0:	RW

		XXXXXXXXXXXXXXXX | Data 0 XXXXXXXXXXXXXXXX | Data 1 1: Data 1 | Data 0 Data 3 | Data 2	
13	SIGN_N	Signed to unsigned or not. If it is 1, the incoming and outgoing audio sample data will be transferred by toggle its most significant bit. 0: Not transfer 1: Do transfer	RW
12	INVALID	Data invalid bit. The data transmitted on SPDIF is valid or not. 0: Valid 1: Invalid	RW
11	SFT_RST	SPDIF FIFO software-reset. Set it to 1 and later it will be cleared by hardware auto. When SFT_RST returns back to 0, the FIFO finish reset. 0: Stop reset 1: Start reset	RW
10	SPDIF_I2S	Choose SPDIF or I2S. 0: I2S 1: SPDIF	
9:2	Reserved	Writing has no effect, read as zero.	R
1	M_TRIG	Trigger interrupt mask. 0: Enabled 1: Masked	RW
0	M_FFUR	FIFO underrun interrupt mask. 0: Enabled 1: Masked	RW

6.2.10 SPDIF State Register (SPSTATE)

The register SPSTATE is used to keep the state of SPDIF.

	SPSTATE																0x10020088													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	Reserved	FIFO_LVL							BUSY	Reserved					F_TRIG	F_FFUR
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15	Reserved	Writing has no effect, read as zero.	R

14:8	FIFO_LVL	FIFO level. The bits indicate the amount of valid data in FIFO.	R
7	BUSY	SPDIF busy bit. 0: SPDIF is not working 1: SPDIF is working	R
6:2	Reserved	Writing has no effect, read as zero.	R
1	F_TRIG	Trigger flag. 0: Not active 1: Active	R
	F_FFUR	FIFO underrun flag. 0: Not active 1: Active	RW

6.2.11 SPDIF Configure 1 Register (SPCFG1)

The register SPCFG1 is used to configure SPDIF.

	SPCFG1																															0x1002008C
Bit	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
	Reserved														INIT_LVL	ZRO_VLD	Reserved		TRIG		SRC_NUM				CH1_NUM				CH2_NUMl			
RST	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

Bits	Name	Description	RW
31:18	Reserved	Writing has no effect, read as zero.	R
17	INIT_LVL	Initial level set bit. 0: SPDIF initial level is low 1: SPDIF initial level is high	
16	ZRO_VLD	The valid bit of channel state is 0 or 1 when play ZERO sample under FIFO underflow. 0: Valid 1: Invalid	RW
15:14	Reserved	Writing has no effect, read as zero.	R
13:12	TRIG	Specify the trigger value of FIFO.	RW
11:8	SRC_NUM	Source number. 0000:Unspecified	RW

TRIG	Description
00	Trigger Value is 4.
01	Trigger Value is 8.
10	Trigger Value is 16.
11	Trigger Value is 32.

		0001~1111:1~15	
7:4	CH1_NUM	Channel 1 number. 0000:Unspecified 0001~1111:A~O	RW
3:0	CH2_NUM	Channel 2 number. 0000:Unspecified 0001~1111:A~O	RW

6.2.12 SPDIF Configure 2 Register (SPCFG2)

The register SPCFG2 is used to configure SPDIF.

	SPCFG2																													0x10020090	
Bit	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
	Reserved		FS				ORG_FRQ				SAMPL_WL			MAX_WL	CLK_ACU		CAT_CODE								CH_MD		Reserved		PRE	COPY_N	AUDIO_N	CON_PRO
RST	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

Bits	Name	Description	RW
31:30	Reserved	Writing has no effect, read as zero.	R
29:26	FS	Sampling frequency. 0000:44.1kHz 0010:48kHz 0011:32kHz 1010:96kHz 1110:192kHz Others: Reference IEC60958-3	RW
25:22	ORG_FRQ	Original sampling frequency. 1111:44.1kHz 1101:48kHz 1100:32kHz 0101:96kHz 0001:192kHz Others: Reference IEC60958-3	RW
21:19	SAMPL_WL	Sample word length. When MAX_WL=1: 001:20 bit 110:21 bit 010:22 bit 100:23 bit	RW

		101:24 bit Others: reserved When MAX_WL=0: 001:16 bit 110:17 bit 010:18 bit 100:19 bit 101:20 bit Others: reserved	
18	MAX_WL	Maximum audio sample word length. 0:20 bit；1:24 bit.	RW
17:16	CLK_ACU	Clock Accuracy of transmitted clock. 00: Level II 01: Level I 10: Level III 11: Interface frame rate not matched to sampling frequency	RW
15:8	CAT_CODE	Category code. Reference IEC60958-3 for full details. 00 indicates “general” mode.	RW
7:6	CH_MD	Channel mode choose bit. 00: Mode 0 01~11: Reserved	RW
5:4	Reserved	Writing has no effect, read as zero.	R
3	PRE	Pre-emphasis set bit. 0: None 1: 15us/15us	RW
2	COPY_N	Copyright set bit. 0: Copyright is asserted 1: Copyright is not asserted	RW
1	AUDIO_N	Linear PCM identification bit. 0: Audio sample word represents linear PCM samples 1: Audio sample word used for other purpose	RW
0	CON_PRO	Consumer mode and professional mode choose bit. 0: Consumer mode 1: Professional mode Professional is not supported in the chip.	RW

6.2.13 SPDIF FIFO Register (SPFIFO)

The register SPCFG1 is used to configure SPDIF.

	SPFIFO																													0x10020094	
Bit	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

	Reserved	DATA
RST	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 0 0 0 0 0 0 0 0

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:0	DATA	FIFO port. When write to it, data is push to the transmit FIFO. Read from it as 0.	W

6.3 Serial Interface Protocol

6.3.1 I2S and MSB-justified serial audio format

Normal I2S and MSB-justified are similar protocols for digitized stereo audio transmitted over a serial path.

The BIT_CLK supplies the serial audio bit rate, the basis for the external CODEC bit-sampling logic. Its frequency is 64 times the audio sampling frequency. Divided by 64, the resulting 8 kHz to 48 kHz or even higher signal signifies timing for left and right serial data samples passing on the serial data paths. This left/right signal is sent to the CODEC on the SYNC pin. Each phase of the left/right signal is accompanied by one serial audio data sample on the data pins SDATA_IN and SDATA_OUT.

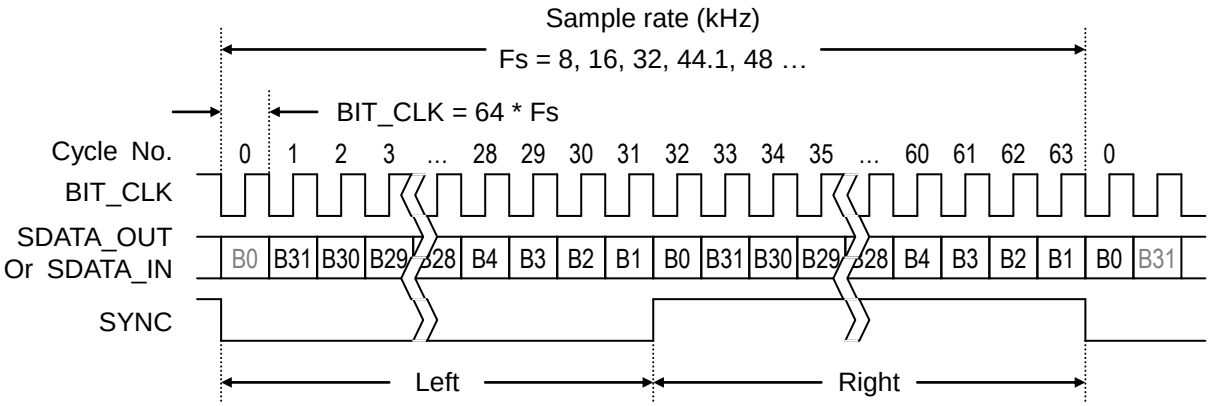


Figure 6-5 I2S data format (A: LR mode)

In the A: LR mode, first send the left channel in a stereo frame. One Left slot and one Right slot make a sample frame. It is the normal mode of I2S.

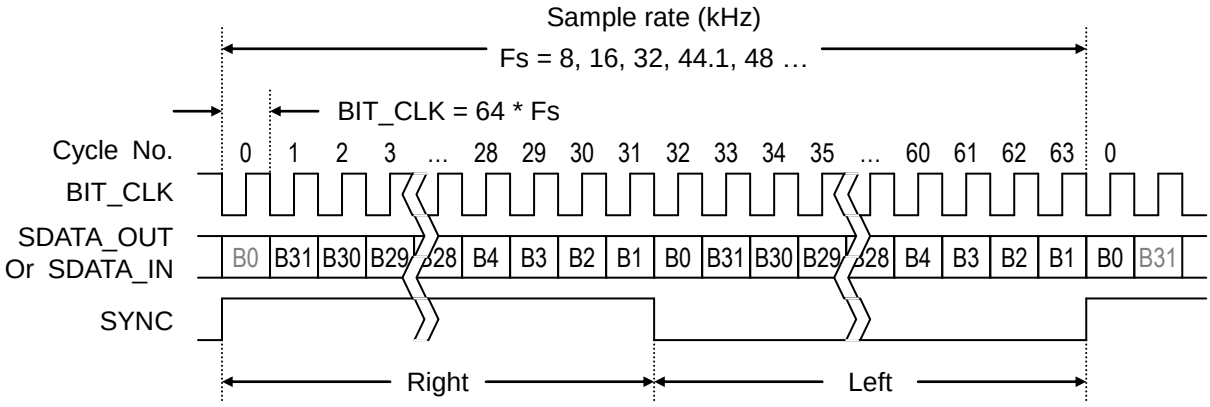


Figure 6-6 I2S data format (B: RL mode)

In the B: RL mode, first send the right channel in a stereo frame. One Right slot and one Left slot make a sample frame. It is used in same CODEC.

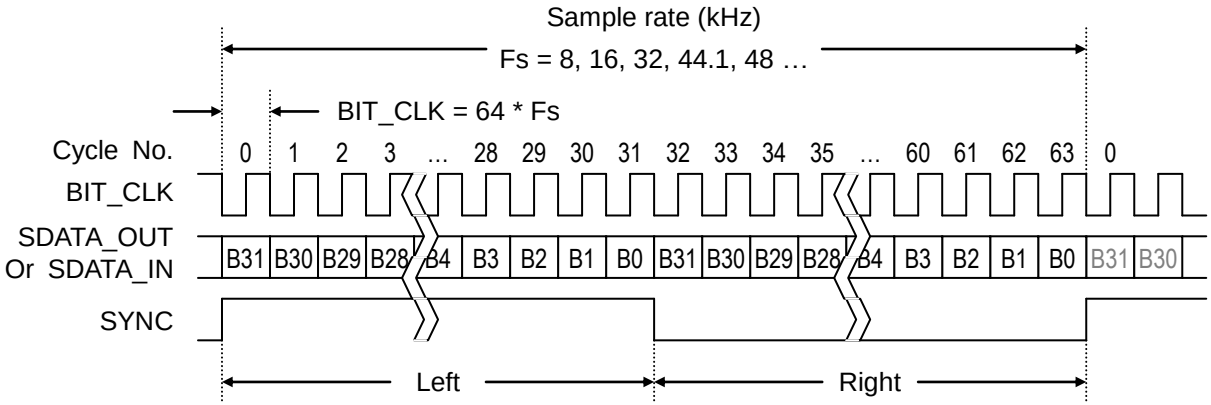


Figure 6-7 MSB-justified data format (C: LR mode)

In the C: LR mode, first send the left channel in a stereo frame. One Left slot and one Right slot make a sample frame. It is the normal mode in MSB-justified.

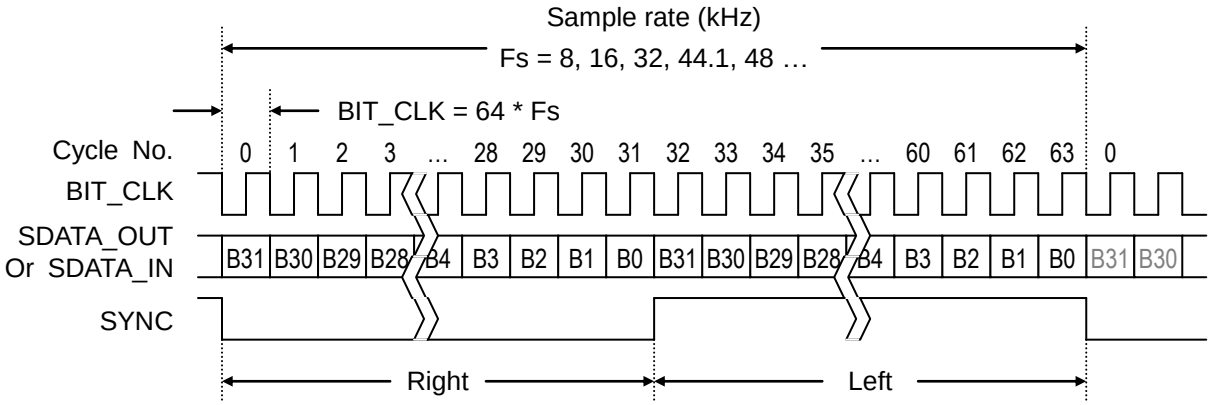


Figure 6-8 MSB-justified data format (D: RL mode)

In the D: RL mode, first send the right channel in a stereo frame. One Right slot and one Left slot make a sample frame.

Figure 6-5 and Figure 6-7 provide timing diagrams that show formats for the normal I2S and MSB-justified modes of operations. Data is sampled on the rising edge of the BIT_CLK and data is sent out on the falling edge of the BIT_CLK.

Data is transmitted and received in frames of 64 BIT_CLK cycles(If BIT_CLK is generated internally). Each frame consists of a left sample and a right sample. Each sample holds 8, 16, 18, 20 or 24 bits of valid data. The LSB other bits of each sample is padded with zeroes.

In the normal I2S mode, the SYNC is low for the left sample and high for the right sample. Also, the MSB of each data sample lags behind the SYNC edges by one BIT_CLK cycle.
In the MSB-justified mode, the SYNC is high for the left sample and low for the right sample. Also, the
MSB of each data sample is aligned with the SYNC edges.

6.3.2 Audio sample data placement in SDATA_IN/SDATA_OUT

The placement of audio sample in incoming/outgoing serial data stream for all formats support in AIC is MSB (Most Significant Bit) justified. Suppose n bit sample composed by

S_{n-1}	S_{n-2}	…	S_2	S_1	S_0

Table 6-3 escribed the how sample data bits are transferred.

Table 6-3 Sample data bit relate to SDATA_IN/SDATA_OUT bit

I2S/MSB-Justified Format					
SDATA IN/OUT	8	16	18	20	24
B31	S7	S15	S17	S19	S23
B30	S6	S14	S16	S18	S22
B29	S5	S13	S15	S17	S21
B28	S4	S12	S14	S16	S20
B27	S3	S11	S13	S15	S19
B26	S2	S10	S12	S14	S18
B25	S1	S9	S11	S13	S17
B24	S0	S8	S10	S12	S16
B23	0	S7	S9	S11	S15
B22	0	S6	S8	S10	S14
B21	0	S5	S7	S9	S13

B20	0	S4	S6	S8	S12
B19	0	S3	S5	S7	S11
B18	0	S2	S4	S6	S10
B17	0	S1	S3	S5	S9
B16	0	S0	S2	S4	S8
B15	0	0	S1	S3	S7
B14	0	0	S0	S2	S6
B13	0	0	0	S1	S5
B12	0	0	0	S0	S4
B11	0	0	0	0	S3
B10	0	0	0	0	S2
B9	0	0	0	0	S1
B8	0	0	0	0	S0
B7~ B0	0	0	0	0	0

If in 16 bits packed mode, the data transferred is the same as the 16 bits normal mode as shown above. But there are two samples in one word.

6.3.3 SPDIF Protocol

SPDIF block format is shown below:

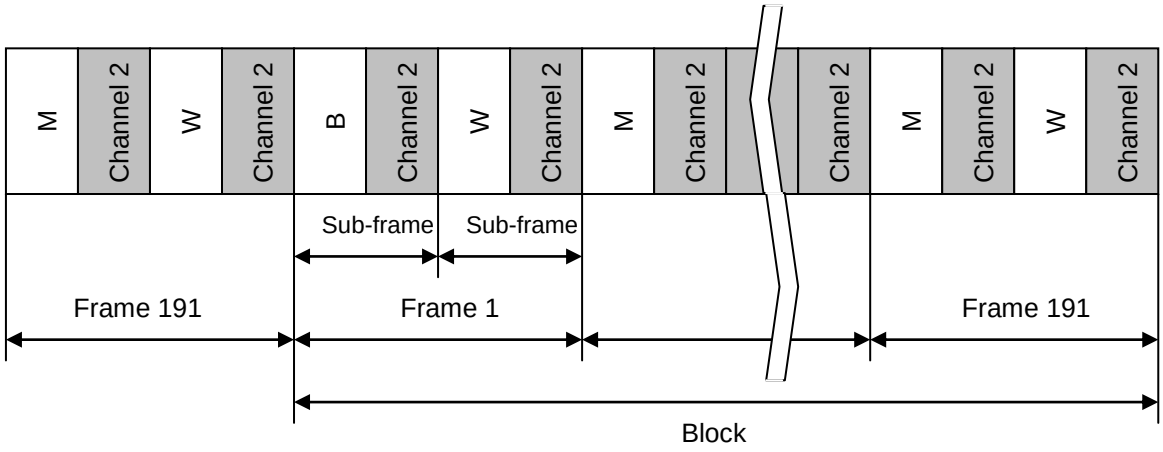


Figure 6-9 Block format

Sub-frame format in PCM mode is shown below:

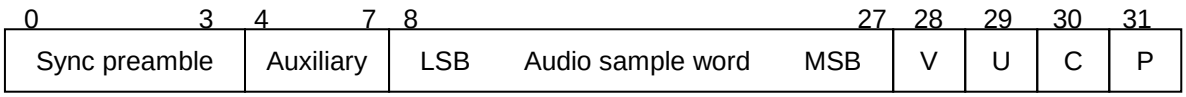


Figure 6-10 Sub-frame format in PCM mode

Sub-frame format in non-PCM mode is shown below:

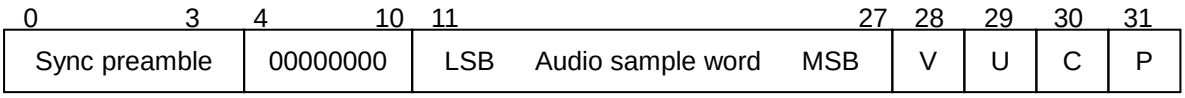


Figure 6-11 Sub-frame format in non-PCM mode

6.4 I2S Operation

The AIC can be accessed either by the processor using programmed I/O instructions or by the DMA controller. The processor uses programmed I/O instructions to access the AIC and can access the following types of data.

- **The AIC memory mapped registers data**—All registers are 32 bits wide and are aligned to word boundaries.
- **AIC controller FIFO data—**An entry is placed into the transmit FIFO by writing to the I2S controller's Serial Audio Data register (AICDR). Writing to AICDR updates a transmit FIFO entry. Reading AICDR flushes out a receive FIFO entry.
- **The external CODEC registers for I2S CODEC—**CODEC registers can be accessed through the L3 bus. The L3 bus operation is emulated by software controlling three GPIO pins.

The DMA controller can only access the FIFOs. Accesses are made through the data registers, as explained in the previous paragraph. The DMA controller responds to the following DMA requests made by the I2S controller:

- The transmit FIFO request is based on the transmit trigger-threshold (AICFR.TFTH) setting. See 6.2.1 for further details regarding AICFR.TFTH.
- The receive FIFO request is based on the receive trigger-threshold (AICFR.RFTH) setting. See 6.2.1 for further details regarding AICFR.RFTH.

Before operation to AIC, you may need to set proper PIN function selection from GPIO using if the pin is shared with GPIO.

6.4.1 Initialization

At power-on or other hardware reset (WDT and etc), AIC is disabled. Software must initiate AIC and the external CODEC after power-on or reset. If errors found in data transferring, or in other places, software must initial AIC and optional, the external CODEC. Here is the initial flow.

1 Select external CODEC (AICFR.ICDC).
2 If I2S/MSB-Justified is selected, select between I2S and MSB-Justified (I2SCR.AMSL).

3 Decide BIT_CLK direction (AICFR.BCKD) and SYNC direction (AICFR.SYNCD).
4 If BIT_CLK is configured as output, BIT_CLK divider I2SDIV.DV must be set to what correspond with the values as shown in Table 6-5. And the clock selection and the divider between PLL clock out and AIC also must be set (CFCR.I2S and I2SCDR in CPM). Enable AIC by write 1 to AICFR.ENB.
5 If it needs to reset AIC registers and flush FIFOs, write 1 to AICFR.RST. If it need only flush FIFOs, write 1 to AICCR.FLUSH. BIT_CLK must exist here and after.
6 In case of external CODEC with I2S/MSB-Justified format, configure I2S/MSB-justified CODEC via the control bus connected to the CODEC, for instance I2C or L3, depends on CODEC. If the resettlement without involving I2S/MSB-justified CODEC or ADC/DAC function changing, this step can be skip.

6.4.2 External CODEC Registers Access Operation

The external audio CODEC can be configured/controlled by its internal registers. To access these registers, an I2S/MSB-justified CODEC usually employs L3 bus, SPI bus, I2C bus or other control bus. The L3 bus operation can be emulated by software by using 3 GPIO pins of the chip.

6.4.3 Audio Replay

Outgoing audio sample data (from AIC to CODEC) is written to AIC transmit FIFO from processor via store instruction or from memory via DMA. AIC then takes the data from the FIFO, serializes it, and sends it over the serial wire SDATA_OUT to an external CODEC or over an internal wire to an internal CODEC.

The audio transmission is enabled automatically when the AIC is enabled by set AICFR.ENB. But all replay data is zero at this time except both of the following conditions are true:

1 AICCR.ERPL must be 1. If AICCR.ERPL is 0, value of zero is send to CODEC even if there are samples in transmit FIFO.
2 At least one audio sample data in the transmit FIFO. If the transmit FIFO is empty, value of zero or last sample depends on AICFR.LSMP, is send to CODEC even if AICCR.ERPL is 1.

Here is the audio replay flow:

1 Configure the CODEC as needed.
2 Configure sample size by AICCR.OSS.
3 Configure sample channels (AICCR.CHANNEL).
4 If sample size is configured 16 bit, select packed or unpacked mode (AICCR.PACK16).
5 If two channels is configured, select the right-channel-first sample data or not (I2SCR.RFIRST).
6 If two channels is configured, select the sample data switched or not (I2SCR.SWLH).
7 Configure sample rate by clock dividers (for I2S/MSB-Justified format with BIT_CLK is provided internally) or by CODEC registers (for BIT_CLK provided by external CODEC).
8 Some other configurations: mono to stereo, endian switch, signed/unsigned data transfer,

transmit FIFO configuration, play ZERO or last sample when TX FIFO under-run, and etc.

9 Write 1 to AICCR.ERPL.
It is suggested that at least a frame of PCM data is pre-filled in the transmit FIFO to prevent FIFO under-run flag (AICSR.TUR).
Fill sample data to the transmit FIFO. Repeat this till finish all sample data. In this procedure, please control the FIFO to make sure no FIFO under-run and other errors happen. When the transmit FIFO under-run, noise or pause may be heard in the audio replay, AICSR.TUR is 1, and if AICCR.ETUR is 1, AIC issues an interrupt. Please reference to 6.4.5 for detail description on FIFO.

10 Waiting for AICSR.TFL change to 0. So that all samples in the transmit FIFO has been replayed, then we can have a clean start up next time.

11 Write 0 to AICCR.ERPL.

6.4.4 Audio Record

Incoming audio sample data (from CODEC to AIC) is received from SDATA_IN (for an external CODEC) serially and converted to parallel word and stored in AIC receive FIFO. Then the data can be taken from the FIFO to processor via load instruction or to memory via DMA.

The audio recording is enabled automatically when the AIC is enabled by set AICFR.ENB. But all received data is discarded at this time except both of the following conditions are true:

1 AICCR.EREC must be 1. If AICCR.EREC is 0, the received data is discarded even if there are rooms in the receive FIFO.

2 At least one room left in the receive FIFO. If the receive FIFO is full, the received data is discarded even if AICCR.EREC is 1.

Here is the audio record flow:

1 Configure the CODEC as needed.

2 Configure sample size by AICCR.ISS.

3 Configure sample rate by clock dividers (for I2S/MSB-Justified format with BIT_CLK is provided internally) or by CODEC registers (for BIT_CLK provided by external CODEC).

4 Some other configurations: signed/unsigned data transfer, receive FIFO configuration, and etc.

5 Write 1 to AICCR.EREC. Make sure there are rooms available in the receive FIFO before set AICCR.EREC. Usually, it should empty the receive FIFO by fetch data from it before set AICCR.EREC.

6 Take sample data form the receive FIFO. Repeat this till the audio finished. In this procedure, please control the FIFO to make sure no FIFO over-run and other errors happen. When the receive FIFO over-run, same recorded audio samples will be lost, AICSR.ROR is 1, and if AICCR.EROR is 1, AIC issues an interrupt. Please reference to 6.4.5 for detail description on FIFO. Write 0 to AICCR.EREC.

7 Take sample data from the receive FIFO until AICSR.RFL change to 0. So that all samples in the receive FIFO has been taken away, then we can have a clean start up next time.

When the receive FIFO is empty, read from it returns zero.

6.4.5 FIFOs operation

AIC has two FIFOs, one for transmit audio sample and one for receive. All AIC played/recorded audio sample data is taken from/send to transmit/receive FIFOs. The RX FIFO is in 24 bits width and 32 entries depth, one entry for keeping one audio sample regardless of the sample size. The RX FIFO is in 32 bits width and 64 entries depth, one entry for keeping one audio sample regardless of the sample size, but in 16 bits packed mode, one entry for keeping two audio samples. AICDR.DATA provides the access point for processor/DMA to write to transmit FIFO and read from receive FIFO. One time access to AICDR.DATA process one sample. The sample data should be put in LSB (Least Significant Bit) in memory or processor registers. For transmitting, bits exceed sample are discarded. For receiving, these bits are set to 0. Figure 6-12 illustrates the FIFOs access.

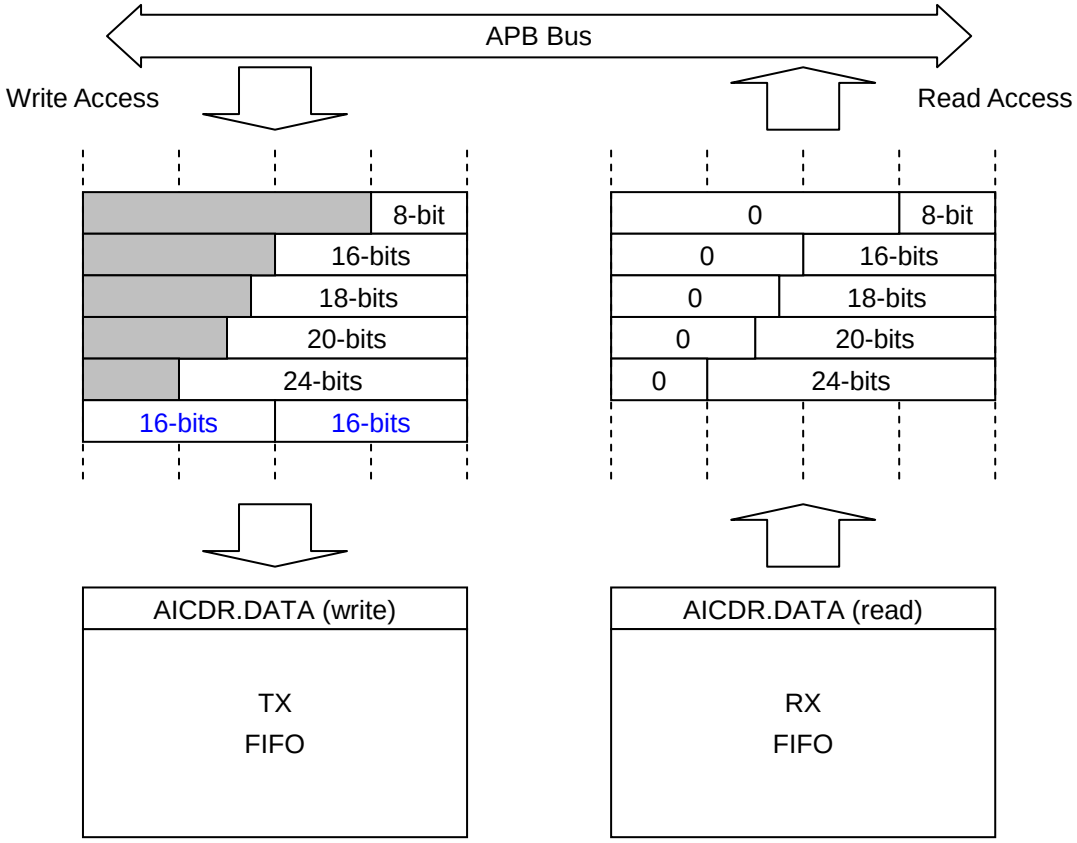


Figure 6-12 Transmitting/Receiving FIFO access via APB Bus

The software and bus initiator must guarantee the right sample placement at the bus.

In case of DMA bus initiator, one 24, 20, 18 bits audio sample must occupies one 32-bits word in memory, so 32-bits width DMA must be used. One 16 bits sample occupies one 16-bits half word in

memory, so 16-bits width DMA must be used. One 8-bits sample occupies one byte in memory, and use 8-bits width DMA except 16bits packed mode. If in 16 bits packed mode, Two 16 bits sample occupies one 32-bits word in memory, so 32-bits width DMA must be used.

In case of processor bus initiator, any type of the audio sample must occupy one CPU general-purpose register at LSB, and read/write from/to AICDR.DATA with 32-bits load/store instruction. When process small sample size, 16-bits or 8-bits, software may need to do the data pack/unpack except 16 bits packed mode. In the 16bits packed mode, the sample data is packed, and two 16 bits audio samples occupy one CPU general-purpose register.

The AICFR.TFTH and AICFR.RFTH are used to set the FIFO level thresholds, which are the trig levels of DMA request and/or FIFO service interrupt. The AICFR.TFTH and AICFR.RFTH should be set to proper value; too small or too big are not good. When AICFR.RFTH is too small, or AICFR.TFTH is too big, the DMA burst length or the number of sample can be processed by processor is too small, which harms the bus or processor efficiency. When AICFR.RFTH is too big or AICFR.TFTH is too small, the bus or the interrupt latency left for under-run/over-run is too small, which may causes replay/record errors.

AICSR.TUR is set to 1 during transmit under-run conditions. If AICCR.ETUR is 1, this can trigger an interrupt. During transmit under-run conditions, zero or last sample is continuously sent out across the serial link. Transmit under-run can occur under the following conditions:

1 Valid transmit data is still available in memory, but the DMA controller/processor starves the transmit FIFO, as it is busy servicing other higher-priority tasks.
2 The DMA controller/processor has transferred all valid data from memory to the transmit FIFO.

AICSR.ROR is set to 1 during receive over-run conditions. If AICCR.EROR is 1, this can trigger an interrupt. During receive over-run conditions, data sent by the CODEC is lost and is not recorded.

When replay/record two channels data, the left channel is default the first data in FIFOs and in the serial link. In 16bits packed mode, could configure that the left channel is the first data or the right channel. By default, the 16 bits LSB is left channel, 16 bits MSB is the right channel. But it also could be switched the Left or the Right channel (I2SCR.SWLH).

6.4.6 Data Flow Control

There are three approaches provided to control/synchronize the audio incoming/outgoing data flow.

6.4.6.1 Polling and Processor Access

AICSR.RFL and AICSR.TFL reflect how many samples exist in receiving and transmitting FIFOs. Through read these register fields, processor can detect when there are samples in receiving FIFO in audio record and then load them from the RX-FIFO, and when there are rooms in transmitting

FIFO in audio replay and then store samples to the TX-FIFO.

Polling approach is in very low efficiency and is not recommended.

6.4.6.2 Interrupt and Processor Access

Set proper values to AICFR.TFTH and AICFR.RFTH, the FIFO interrupts trig thresholds. Set AICCR.ETFS and/or AICCR.ERFS to 1 to enable transmitting and/or receiving FIFO level trigger interrupts. When the interrupt found, it means there are rooms or samples in the TX or RX FIFO, and processor can store or load samples to or from the FIFO.

Interrupt approach is more efficient than polling approach.

6.4.6.3 DMA Access

Audio data is real time stream, though it is in low data bandwidth, usually less than 1.2Mbps. DMA approach is the most efficient and is the recommended approach.

To enable DMA operation, set AICCR.TDMS and AICCR.RDMS to 1 for transmit and receive respectively. It also needs to allocate two channels in DMA controller for data transmitting and receiving respectively. Please reference to the processor's DMA controller spec for the details.

The AICFR.TFTH and AICFR.RFTH are used to set the transmitting and receiving FIFO level thresholds, which determine the issuing of DMA request to DMA controller. To respond the request, DMAC initiator and controls the data movement between memory and TX/RX FIFO.

6.4.7 Audio Samples format

6.4.7.1 16 bits packed mode

One channel (mono) mode and two channels (stereo) mode:

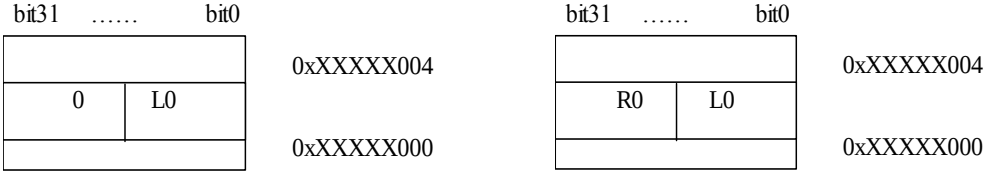


Figure 6-13 One channel (Left) and Two channels (right) mode (16 bits packed mode)

Four channels mode and six channels mode:

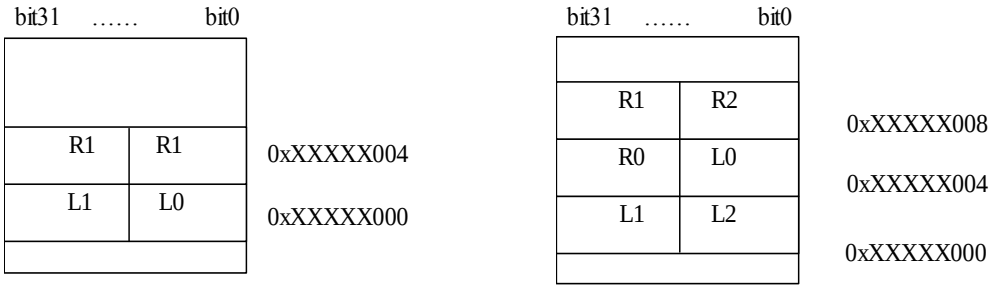


Figure 6-14 Four channels (Left) and Six channels (right) mode (16 bits packed mode)

Eight channels mode:

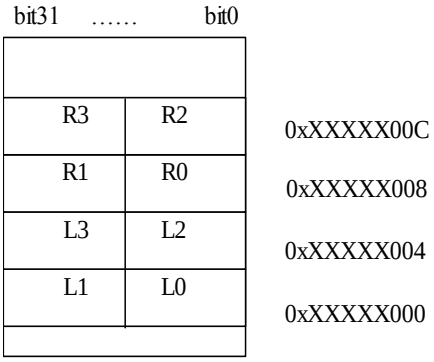


Figure 6-15 Eight channels mode (16 bits packed mode)

6.4.7.2 Normal mode.

One channel (Mono) and two channels (stereo) mode:

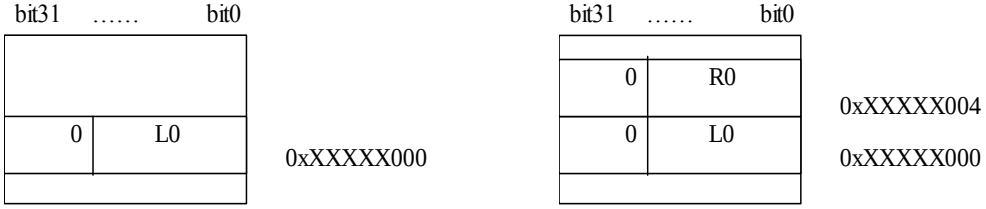


Figure 6-16 One channel (Left) and Two channels (right) mode

Four channels mode and six channels mode:

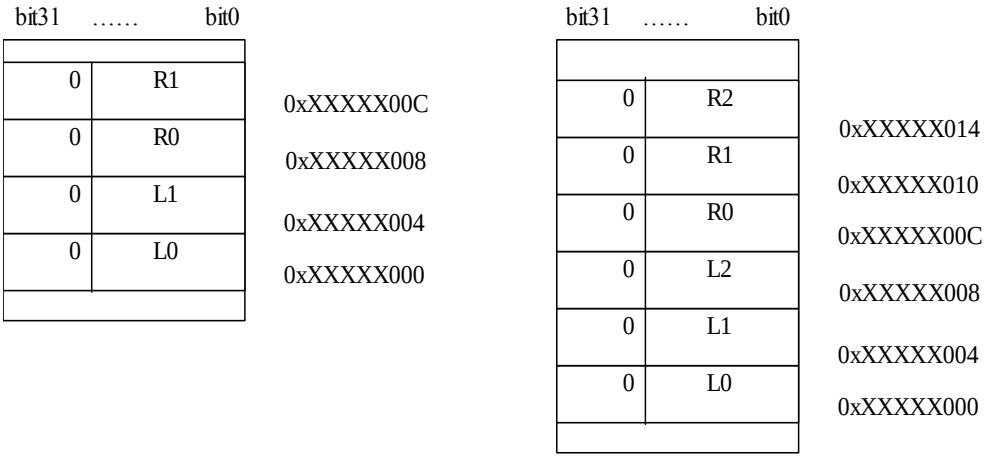


Figure 6-17 Four channels (Left) and Six channels (right) mode

Eight channel mode:

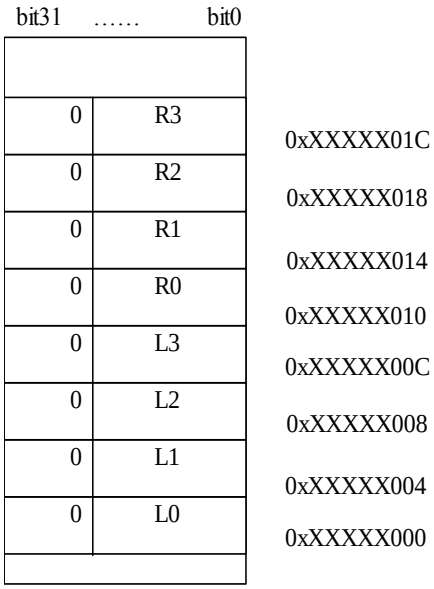


Figure 6-18 Eight channels mode

6.4.8 Serial Audio Clocks and Sampling Frequencies

Following are for BIT_CLK/SYS_CLK configuration in I2S/MSB-Justified format with external CODEC.

The BIT_CLK is the rate at which audio data bits enter or leave the AIC. BIT_CLK can be supplied either by the CODEC or an internally PLL. If it is supplied internally, BIT_CLK is configured as output

pins, and is supplied out to the CODEC. If BIT_CLK is supplied by the CODEC, then it is configured as an input pin. Register bit AICFR.BCKD is used to select BIT_CLK direction.

The audio sampling frequency is the frequency of the SYNC signal, which must be 1/64 of BIT_CLK, $f_{BIT\_CLK} = 64\ f_S$.
SYS_CLK is only for CODEC. It usually takes one of the two roles, as CODEC master clock input or as CODEC over-sampling clock input. If SYS_CLK roles as CODEC master clock input, it usually should be set to a fixed frequency according to CODEC requirement but independent to audio sample rate. In this case, usually there is a PLL in the CODEC and CODEC roles master mode. See Figure 6-2 for the interface diagram. This is the recommended AIC CODEC system configuration.

If SYS_CLK roles as CODEC over-sampling clock, its frequency is usually 4, 6, 8 or 12 times of BIT_CLK frequency, which are 256, 384, 512 and 768 times of audio sample rates. Table 6-4 ists the relation between sample rate, BIT_CLK and SYS_CLK frequencies.

Table 6-4 Audio Sampling rate, BIT_CLK and SYS_CLK frequencies

Sample Rate f_S (kHz)	**BIT_CLK (MHz) $f_{BIT\_CLK} = 64\ f_S$**	**SYS_CLK (MHz)**			
		256 f_S	**384 f_S**	**512 f_S**	**768 f_S**
48	3.072	12.288	18.432	24.576	36.864
44.1	2.8224	11.2896	16.9344	22.5792	33.8688
32	2.048	8.192	12.288	16.384	24.576
24	1.536	6.144	9.216	12.288	18.432
22.05	1.4112	5.6448	8.4672	11.2896	16.9344
16	1.024	4.096	6.144	8.192	12.288
11.025	0.7056	2.8224	4.2336	5.6448	8.4672
8	0.512	2.048	3.072	4.096	6.144

In this processor, SYS_CLK can be selected from EXCLK or generated by dividing the PLL output clock in a CPM divider controlled by I2SCDR. If BIT_CLK is chosen as an output, another divider in AIC is used to divide SYS_CLK for it.

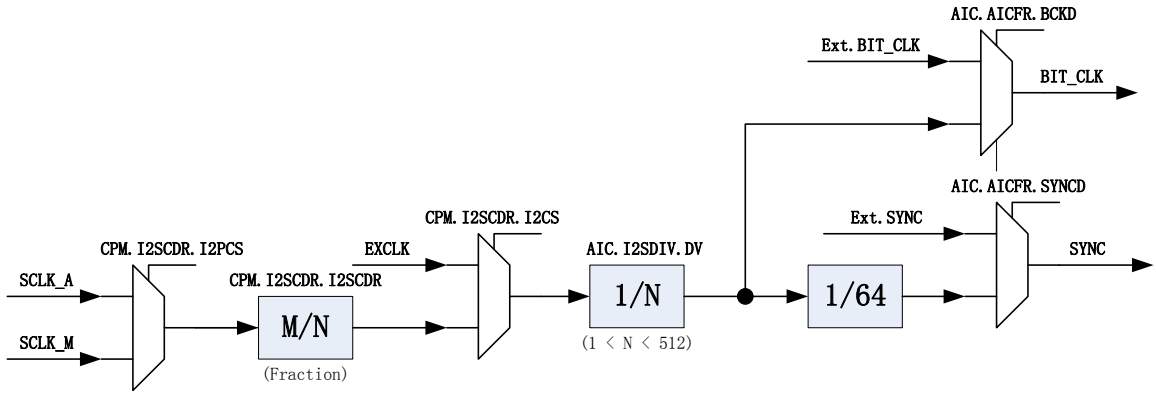


Figure 6-19 SYS_CLK, BIT_CLK and SYNC generation scheme

Note: SCLK_A and SCLK_M please refer to CPM.

The setting of I2SDIV.DV is shown in Table 6-5.

Table 6-5 BIT_CLK divider setting

I2SDIV.DV	f_{SYS_CLK}	f_{BIT_CLK}	f_{SYS_CLK} / f_{BIT_CLK}
0x0	64fs	64fs	1
0x1	128 f_S	64 f_S	2
0x2	196 f_S	64 f_S	3
0x3	256 f_S	64 f_S	4
0x5	384 f_S	64 f_S	6
0x7	512 f_S	64 f_S	8
0xB	768 f_S	64 f_S	12

As we observe in Table 6-4, if SYS_CLK is taken as over-sampling clock by CODEC, the common multiple of all SYS_CLK frequencies is much bigger than the PLL output clock frequency. To generate all different SYS_CLK frequencies, one approach is change PLL frequency according to sample rate. This is not realistic, since frequently change PLL frequency during normal operation is not recommended.

Another approach is to found some approximate common multiples of all SYS_CLK frequencies according to the fact that there tolerance in audio sample rate. Take f_{SYS_CLK} = 256 f_S, Table 6-6 list most frequencies, which are less than 400MHz, with relatively small sample rate errors. It is suggested to set PLL frequency as close to the frequencies listed as possible, then use clock dividers to generate different SYS_CLK/BIT_CLK for different sample rate.

Table 6-6 Approximate common multiple of SYS_CLK for all sample rates

Approximate Common Frequency (MHz)	Max Error Caused in Audio Sample Rate (%)
123.53	0.53
147.11	0.24
170.68	0.79
235.5	0.87
247.06	0.53
270.64	0.11
280.56	0.73
294.22	0.24
305.14	0.67
317.79	0.53
329.57	0.66
341.35	0.79
347	0.85
353.13	0.90
358.79	0.69
370.59	0.53
382.96	0.54
394.17	0.24

Take PLL = 270.64 MHz as an example, Table 6-7 lists the divider settings for various sample rates.

Table 6-7 CPM/AIC clock divider setting for various sampling rate if PLL = 270.64MHz

Sample Rate (kHz)	I2SCDR	I2SDIV.DV	Sample Rate Error (%)
48	1	11	0.11
44.1	1	12	-0.11
32	0	33	0.11
24	1	22	0.11
22.05	1	24	-0.11
16	1	33	0.11
12	1	44	0.11
11.025	1	48	-0.11
8	1	66	0.11

For an EXCLK clock frequency, try to generate PLL frequencies as close to the frequencies listed in Table 6-6 as possible. Table 6-8 lists the PLL parameters and audio sample errors at different PLL frequencies for EXCLK at 12MHz.

Table 6-8 PLL parameters and audio sample errors for EXCLK=12MHz

PLL			Max Sample Rate Error
M	N	Freq (MHz)	
103	10	123.6	0.59%
49	4	147	0.31%
128	9	170.67	0.79%
157	8	235.5	0.87%
103	5	247.2	0.59%
65	3	260	0.82%
45	2	270	0.35%
203	9	270.67	0.12%
113	5	271.2	0.32%
187	8	280.5	0.75%
237	10	284.4	0.81%
49	2	294	0.31%
178	7	305.14	0.67%
53	2	318	0.60%
302	11	329.45	0.70%
256	9	341.33	0.79%
318	11	346.91	0.88%
206	7	353.14	0.90%
299	10	358.8	0.69%
247	8	370.5	0.55%
351	11	382.91	0.55%
230	7	394.29	0.27%

The BIT_CLK should be stopped temporary when change the divider settings, or when change BIT_CLK source (from external), to prevent clock glitch. Register I2SCR.STPBK is provided to assist the task. When I2SCR.STPBK = 1, BIT_CLK is disabled no matter whether it is generated internally or inputted from the external source. The operation flow is described in following.

1 Stop all replay/record by clear AICCR.ERPL and AICCR.EREC.
2 Polling I2SSR.BSY till it is 0.
3 Stop the BIT_CLK by write 1 to I2SCR.STPBK.
4 Operations concerning BIT_CLK.
5 Resume the BIT_CLK by write 0 to I2SCR.STPBK.

6.4.9 Interrupts

The following status bits, if enabled, interrupt the processor:

- Receive FIFO Service (AICSR.RFS). It's also DMA Request.
- Transmit FIFO Service (AICSR.TFS). It's also DMA Request.
- Transmit Under-Run (AICSR.TUR).

- Receive Over-Run (AICSR.ROR).

For further details, see the corresponding register description sections.

6.5 SPDIF Guide

6.5.1 Set SPDIF clock frequency

Set SPDIF clock frequency is as same as i2s clock.

6.5.2 PCM audio mode operation (Reference IEC60958)

1 Set SPCFG1 and SPCFG2 to configure SPDIF transmitter.
   a Set SPCFG2.CON_PRO to 0 to choose consumer mode.
   b Set SPCFG2. AUDIO_N to 0 to choose linear PCM audio data mode.
   c Set SPCFG1.XXX to configure SPDIF.
   d Set SPCFG2.XXX to configure SPDIF.
2 Set SPCTRL.DMA_EN to choose DMA mode or CPU mode.
3 Set SPCTRL.SIGN_N to choose whether to transfer the most significant bit by toggle or not.
4 Set SPCTRL. SFT_RST to 1 reset FIFO.
5 Wait SPCTRL. SFT_RST set to be set 0 by hardware.
6 Set SPCTRL.M_TRIG and SPCTRL.M_FFUR to enable or disable the interrupt.
7 Set SPCTRL.INVALID 1 or 0 to set the V bit of sub-frame.
8 Set SPENA.SPEN to 1 to Enable SPDIF to transmitter.

6.5.3 Non-PCM mode operation (Reference IEC61937)

1 Set SPCFG1 and SPCFG2 to configure SPDIF transmitter.
   a Set SPCFG2.CON_PRO to 0 to choose consumer mode.
   b Set SPCFG2.AUDIO_N to 1 to choose non-PCM mode.
   c Set SPCFG1.SRC_NUM to 0.
   d Set SPCFG1.CH1_NUM to 0.
   e Set SPCFG1.CH2_NUM to 0.
   f Set SPCFG2.PRE to 0.
   g Set SPCFG2.CH_MD to 0.
   h Set SPCFG2.ORG_FRQ to 0.
   i Set SPCFG2.SAMPL_WL to 0.
   j Set SPCFG2.MAX_WL to 0.
   k Set SPCFG1.XXX to configure SPDIF.
   l Set SPCFG2.XXX to configure SPDIF.
2 Set SPCTRL.DMA_EN to choose DMA mode or CPU mode.
3 Set SPCTRL.SIGN_N to choose whether to transfer the most significant bit by toggle or not.
4 Set SPCTRL. SFT_RST to 1 reset FIFO.

5 Wait SPCTRL. SFT_RST to be set to 0 by hardware.
6 Set SPCTRL.M_TRIG and SPCTRL.M_FFUR to enable or disable the interrupt.
7 Set SPCTRL.INVALID 1 or 0 to set the V bit of sub-frame.
8 Set SPENA.SPEN to 1 to Enable SPDIF to transmitter.

6.5.4 Disable operation

1 Set SPENA.SPEN to 0 to disable SPDIF to transmitter.
2 Wait SPSTATE.BUSY to be set to 0 by hardware.
3 You can do other operation.

7 PCM Interface

7.1 Overview

The PCM has the following features:

- Data starts with the frame PCMSYN or one PCMCLK later
- Support three modes of operation for PCM
  - Short frame sync mode
  - Long frame sync mode
  - Multi-slot mode
- Data is transferred and received with the MSB first
- Support master mode and slave mode
- The PCM serial output data, PCMDOUT, is clocked out using the rising edge of the PCMSCLK
- The PCM serial input data, PCMDIN, is clocked in on the falling edge of the PCMSCLK
- 8/16 bit sample data sizes supported
- DMA transfer mode supported
- Two FIFOs for transmit and receive respectively with 16 samples capacity in every direction

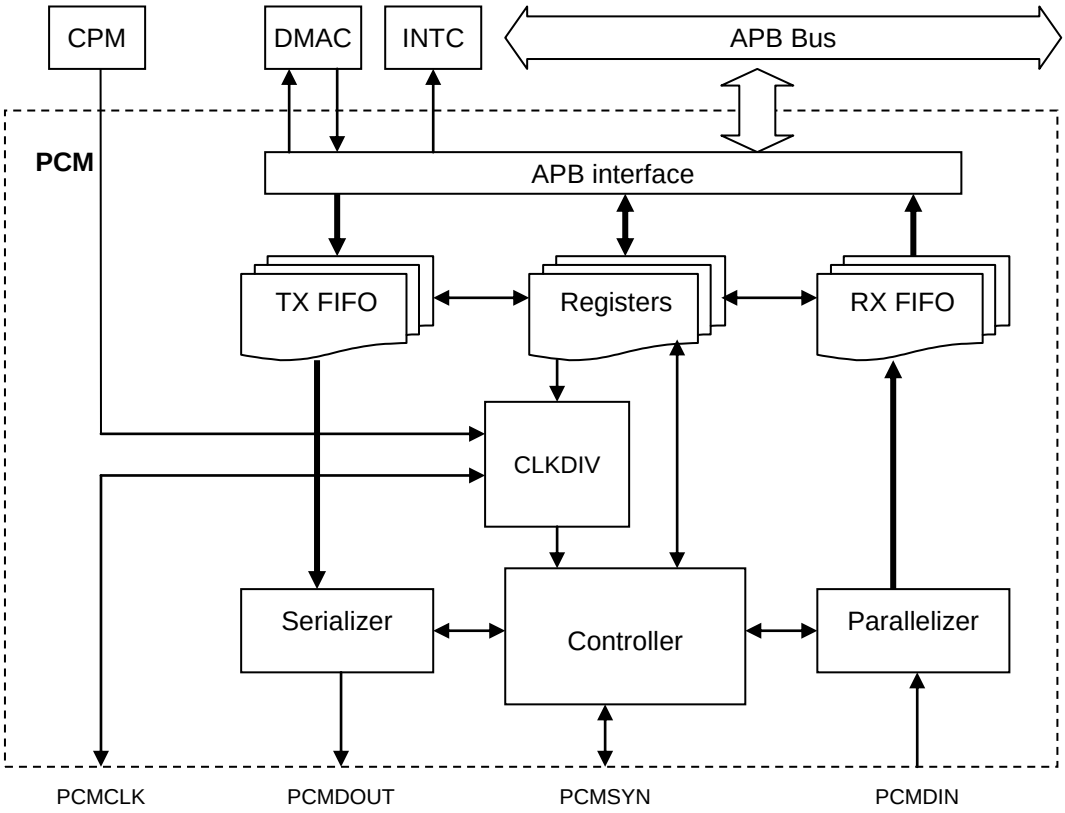
7.2 Pin Description

There are all 4 pins used to connect between PCM interface and an external device. They are listed and described in Table 7-1.

Table 7-1 PCM Interface Pins Description

Name	I/O	Description
PCMCLK	Input / Output	PCM Serial clock Line signal input/output
PCMSYN	Input / Output	PCM sync signal input/output
PCMDOUT	Output	PCM Serial data output
PCMDIN	Input	PCM Serial data input

7.3 Block Diagram



7.4 Registers

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute
   R - Read only
   W - Write only
   RW - read and write
   RCW - read and write, but clear to 0 by read
   RSW - read and write, but set to 1 by read
   RWC - read and write, clear to 0 by write 1, write 0 has no effect
   RWS - read and write, set to 1 by write 1, write 0 has no effect
   RC - read only, and clear to 0 by read
   RS - read only, and set to 1 by read
   SPEC - special access method, relate to its description
3. Reset Value
   1 - reset to 1

0 - reset to 0

? - value unknown after reset

7.4.1 Registers Memory Map

Table 7-2 Registers Memory Map-Address Base

Name	Base	Description
PCM0	0x10071000	Address base of DMIC Controller.

Table 7-3 PCM Registers Description

Name	Description	Reset Value	Address offset	Access Size
PCMCTL0	PCM Control Register	0x00000000	0x000	32
PCMCFG0	PCM Configure Register	0x00000110	0x004	32
PCMDP0	PCM FIFO Data Port Register	0x00000000	0x008	32
PCMINTC0	PCM Interrupt Control Register	0x00000000	0x00c	32
PCMINTS0	PCM Interrupt Status Register	0x00000100	0x010	32
PCMDIV0	PCM Clock Divide Register	0x00000001	0x014	32

7.4.2 Register Description

7.4.2.1 PCM Control Register (PCMCTL)

	PCMCTL0																						BASE + 0x000									
Bit	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
	Reserved																						ERDMA	ETDMA	LSMP	ERPL	EREC	FLUSH	RST	SYNEN	CLKEN	PCMEN
RST	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

Bits	Name	Description	RW
31:10	Reserved	Writing has no effect, read as zero.	R
9	ERDMA	Receive DMA Enable. This bit is used to enable or disable the DMA during receiving audio data. ERDMA: 0 - Disabled. 1 - Enabled.	RW
8	ETDMA	Transmit DMA Enable. This bit is used to enable or disable the DMA during transmit audio data.	RW

			ETDMA	**Transmit DMA**
			0	Disabled.
			1	Enabled.
7	LSMP	Select between play last sample or play ZERO sample in TX FIFO underflow. ZERO sample means sample value is zero. **LSMP** / **CODEC used** 0 / Play ZERO sample when TX FIFO underflow. 1 / Play last sample when TX FIFO underflow.		RW
6	ERPL	Enable Playing Back function. This bit is used to disable or enable the audio sample data transmitting. **ERPL** / **Description** 0 / PCM Playing Back Function is Disabled. 1 / PCM Playing Back Function is Enabled.		RW
5	EREC	Enable Recording Function. This bit is used to disable or enable the audio sample data receiving. **EREC** / **Description** 0 / PCM Recording Function is Disabled. 1 / PCM Recording Function is Enabled.		RW
4	FLUSH	FIFO Flush. Write 1 to this bit flush transmit/receive FIFOs to empty. Writing 0 to this bit has no effect and this bit is always reading 0.		W
3	RST	Reset PCM. Write 1 to this bit reset PCM registers and FIFOs. Writing 0 to this bit has no effect and this bit is always reading 0.		W
2	Reserved	Writing has no effect, read as zero.		R
1	CLKEN	Enable the serial clock division logic. Must be HIGH for the PCM to operate.		RW
0	PCMEN	Enable PCM function. This bit is used to enable or disable the PCM function. **PCMENB** / **Description** 0 / Disable PCM Controller. 1 / Enable PCM Controller.		RW

7.4.2.2 PCM Configuration Register (PCMCFG)

	PCMCFG0																BASE + 0x004													
Bit	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
	Reserved																	SLOT		ISS	OSS	IMSBPOS	OMSBPOS	RFTH				TFTH				PCMMOD
RST	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	1	0	0	0	0

<table>
<tr><th>Bits</th><th>Name</th><th>Description</th><th>RW</th></tr>
<tr><td>31:15</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>14:13</td><td>SLOT</td><td>Controls the amount of valid PCM timeslot in one PCMSYN frame.
<table>
<tr><th>SLOT</th><th>Number of slot in one frame</th></tr>
<tr><td>00</td><td>1 slot</td></tr>
<tr><td>01</td><td>2 slot</td></tr>
<tr><td>10</td><td>3 slot</td></tr>
<tr><td>11</td><td>4 slot</td></tr>
</table></td><td>RW</td></tr>
<tr><td>12</td><td>ISS</td><td>Input Sample Size. These bits reflect input sample data size to memory or register. The data sizes supported are: 8/16bits. The sample data is LSB-justified in memory/register.
<table>
<tr><th>ISS</th><th>Sample Size</th></tr>
<tr><td>0</td><td>8 bit</td></tr>
<tr><td>1</td><td>16 bit</td></tr>
</table></td><td>RW</td></tr>
<tr><td>11</td><td>OSS</td><td>Output Sample Size. These bits reflect output sample data size from memory or register. The data sizes supported are: 8/16 bits. The sample data is LSB-justified in memory/register.
<table>
<tr><th>OSS</th><th>Sample Size</th></tr>
<tr><td>0</td><td>8 bit</td></tr>
<tr><td>1</td><td>16 bit</td></tr>
</table></td><td>RW</td></tr>
<tr><td>10</td><td>IMSBPOS</td><td>Controls the position of the MSB bit in the serial input stream relative to the PCMSYN signal.
0: MSB is captured on the falling edge of PCMCLK during the same cycle that PCMSYNC is high
1: MSB is captured on the falling edge of PCMCLK during the cycle after the PCMSYNC is high</td><td>RW</td></tr>
<tr><td>9</td><td>OMSBPOS</td><td>Controls the position of the MSB bit in the serial output stream relative to the PCMSYN signal.
0: MSB sent during the same clock that PCMSYN is high
1: MSB sent on the next PCMSCLK cycle after PCMSYNC is high</td><td>RW</td></tr>
<tr><td>8:5</td><td>RFTH</td><td>Receive FIFO threshold for interrupt or DMA request. Determines when the RFS flags go active for the RXFIFO. When the sample number in receive FIFO, indicated by PCMINTS.RFL, is great than the threshold value, PCMINTS.RFS is set. Larger RFTH value provides lower DMA/interrupt request frequency but have more risk to involve receive FIFO overflow. The optimum value is system dependent.</td><td>RW</td></tr>
<tr><td>4:1</td><td>TFTH</td><td>Transmit FIFO threshold for interrupt or DMA request. Determines when the TFS flags go active for the TXFIFO. When the sample number in transmit FIFO, indicated by PCMINTS.TFL, is less than the threshold value, PCMINTS.TFS is set. Smaller TFTH value provides lower DMA/interrupt request frequency but have more risk</td><td>RW</td></tr>
</table>

		to involve transmit FIFO underflow. The optimum value is system dependent.	
0	PCMMOD	PCM mode select. 0: Master mode; 1: Slave mode.	RW

7.4.2.3 PCM FIFO DATA PORT REGISTER (PCMDP)

	PCMDP0																BASE + 0x008															
Bit	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
	DATA																															
RST	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

Bits	Name	Description	RW
31:0	DATA	FIFO port. When write to it, data is push to the transmit FIFO. When read from it, data is pop from the receiving FIFO.	RW

7.4.2.4 PCM INTERRUPT CONTROL REGISTER (PCMINTC)

	PCMINTCR0																BASE + 0x00c															
Bit	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
	Reserved																												ETFS	ETUR	ERFS	EROR
RST	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

Bits	Name	Description	RW
31:4	Reserved	Writing has no effect, read as zero.	R
3	ETFS	Enable TFS Interrupt. This bit is used to control the TFS interrupt enable or disable. **ETFS** / **TFS Interrupt** 0 / Disabled. 1 / Enabled.	RW
2	ETUR	Enable TUR Interrupt. This bit is used to control the TUR interrupt enable or disable. **ETUR** / **TUR Interrupt** 0 / Disabled. 1 / Enabled.	RW

1	ERFS	Enable RFS Interrupt. This bit is used to control the RFS interrupt enable or disable. **ERFS** / **RFS Interrupt** 0 / Disabled. 1 / Enabled.	RW
0	EROR	Enable ROR Interrupt. This bit is used to control the ROR interrupt enable or disable. **EROR** / **ROR Interrupt** 0 / Disabled. 1 / Enabled.	RW

7.4.2.5 PCM INTERRUPT STATUS REGISTER (PCMINTS)

PCMINTS0 — **BASE + 0x010**

Bit	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
	Reserved																	RSTS	TFL						TFS	TUR	RFL				RFS	ROR
RST	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

Bits	Name	Description	RW
31:15	Reserved	Writing has no effect, read as zero.	R
14	RSTS	Soft reset / flush state. 0: Nothing / reset or flush operation has completed 1: reset or flush operation has not completed	R
13:9	TFL	Transmit FIFO Level. The bits indicate the amount of valid PCM data in Transmit FIFO.	R
8	TFS	Transmit FIFO Service Request. This bit indicates that transmit FIFO level exceeds TFL threshold which is controlled by PCMCFG.TFTH When TFS is 1, it may trigger interrupt or DMA request depends on the interrupt enable and DMA setting. **TFS** / **Description** 0 / Transmit FIFO level exceeds TFL threshold. 1 / Transmit FIFO level at or below TFL threshold.	RW
7	TUR	Transmit FIFO Under Run. This bit indicates that transmit FIFO has or has not experienced an under-run. **TUR** / **Description** 0 / When read, indicates under-run has not been found.	RW

				When write, clear itself.	
			1	When read, indicates data has even been read from empty transmit FIFO.	
				When write, not effects.	
6:2	RFL	Receive FIFO Level. The bits indicate the amount of valid PCM data in Receive FIFO.			R
1	RFS	Receive FIFO Service Request. This bit indicates that receive FIFO level is or not below RFL threshold which is controlled by PCMCFG.RFTH. When RFS is 1, it may trigger interrupt or DMA request depends on the interrupt enable and DMA setting.			RW

RFS	Description
0	Receive FIFO level below RFL threshold.
1	Receive FIFO level at or above RFL threshold.

0	ROR	Receive FIFO Over Run. This bit indicates that receive FIFO has or has not experienced an overrun.			RW

ROR	Description
0	When read, indicates over-run has not been found.
	When write, clear itself.
1	When read, indicates data has even been written to full receive FIFO.
	When write, not effects.

7.4.2.6 PCM CLOCK DIVIDE REGISTER (PCMDIV)

	PCMDIV0																BASE + 0x014													
Bit	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
	Reserved																SYNL					SYNDIV					CLKDIV					
RST	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

Bits	Name	Description	RW
31:27	Reserved	Writing has no effect, read as zero.	R
16:11	SYNL	Controls the length that the PCMSYN based upon the PCMCLK. The length of PCMSYN = (SYNL + 1) * PCMCLK cycle.	RW
10:6	SYNDIV	Controls the frequency of the PCMSYN signal based upon the PCMCLK. PCMSYN = PCMCLK / 8 (SYNDIV + 1).	RW
5:0	CLKDIV	PCMCLK clock divider value minus 1. Controls the divider used to create the PCMCLK based upon the CPM_PCM_SYSCLK. PCMCLK = CPM_PCM_SYSCLK / (CLKDIV + 1).	RW

7.5 PCM Interface Timing

The following figures show the timing relationship for the PCM transfers, Note in all cases. In master mode, the PCMCLK is derived from dividing the input clock, CPM_PCM_SYSCLK, and the PCMSYN is divided depended on the PCMCLK. In slave mode, the PCMCLK and PCMSYN are input from the external device. Data is sampled on the falling edge of the PCMCLK and sent out on the rising edge of the PCMCLK. The PCMSYN signal determines when the next data sample is to be transferred between the controller and the external device. Also, the PCMSYN signal as seen in the figure can be one bit time or a long bit time controlled by PCMDIV.SYNL. The PCMSYN frequency controlled by PCMDIV.SYNDIV is usually the sample rate. There are some variations controlled by PCMCFG.ISS, PCMCFG.OSS and PCMCFG.SLOT to accommodate 8 / 16bit sample sizes and multi-slot transmission.

7.5.1 Short Frame SYN

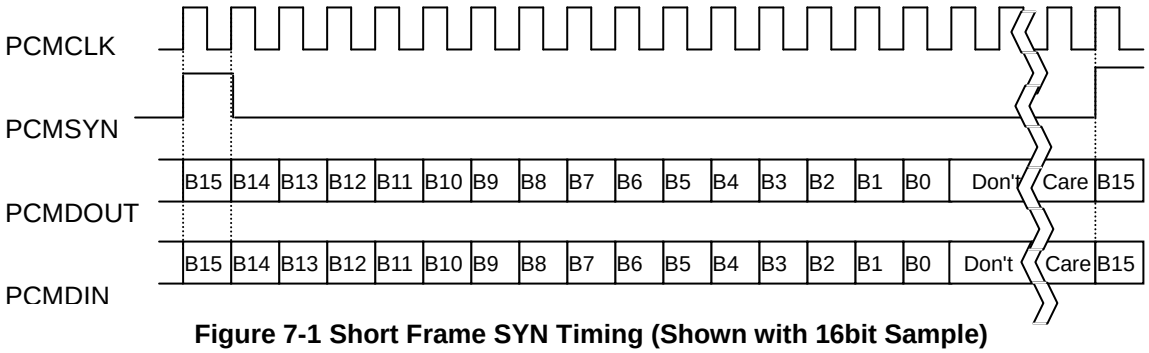


Figure 7-1 Short Frame SYN Timing (Shown with 16bit Sample)

NOTE: Figure 7-1 shows a PCM transfer with the MSB configured to be coincident with the PCMSYN.

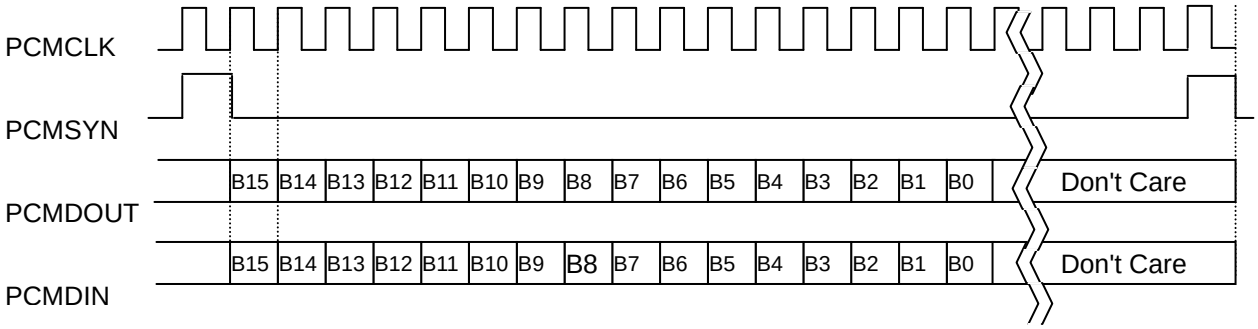


Figure 7-2 Short Frame SYN Timing (Shown with 16bit Sample)

NOTE: Figure 7-2 shows a PCM transfer with the MSB configured one shift clock after the PCMSYN.

7.5.2 Long Frame SYN

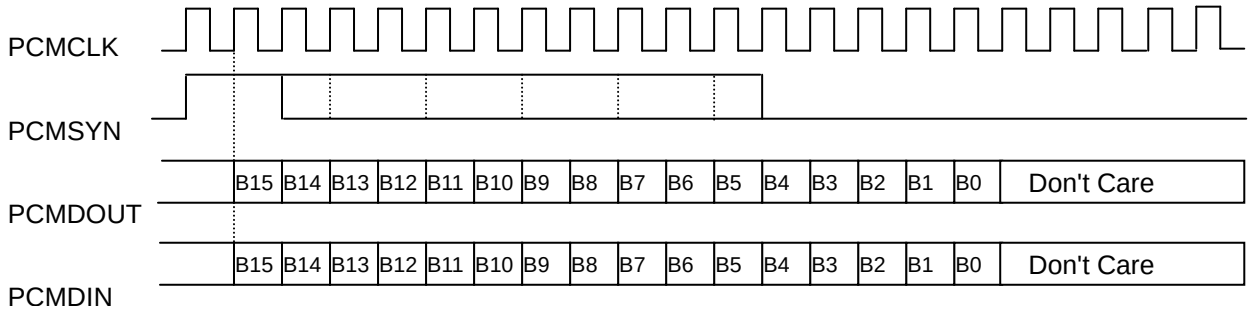


Figure 7-3 Long Frame SYN Timing (Shown with 16bit Sample)

NOTE: Figure 7-3 shows a PCM transfer with the MSB configured one shift clock after the PCMSYN.

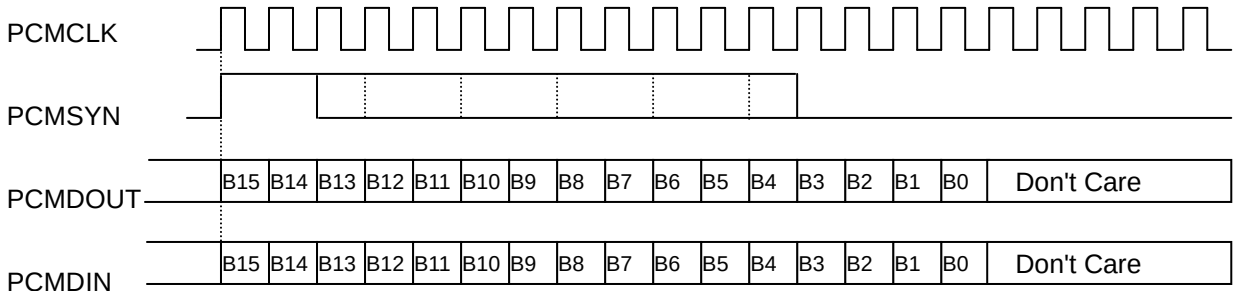


Figure 7-4 Long Frame SYN Timing (Shown with 16bit Sample)

NOTE: Figure 7-4 shows a PCM transfer with the MSB configured to be coincident with the PCMSYN.

7.5.3 Multi-Slot Operation

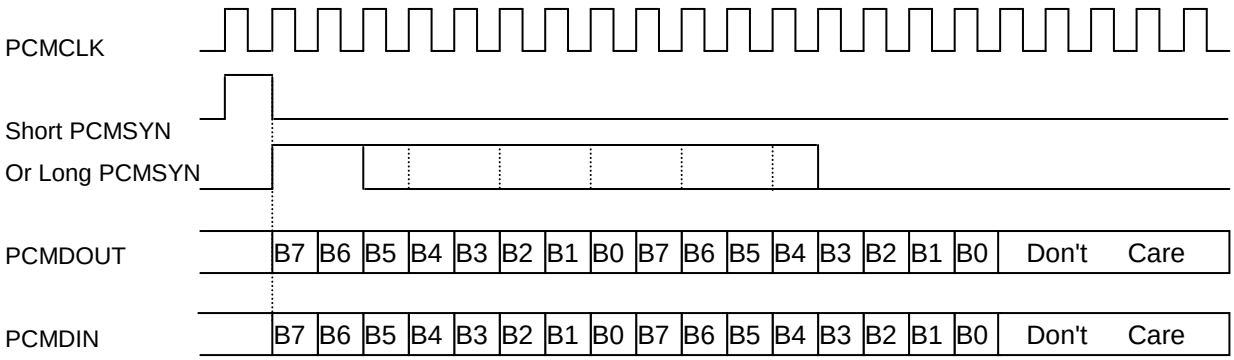
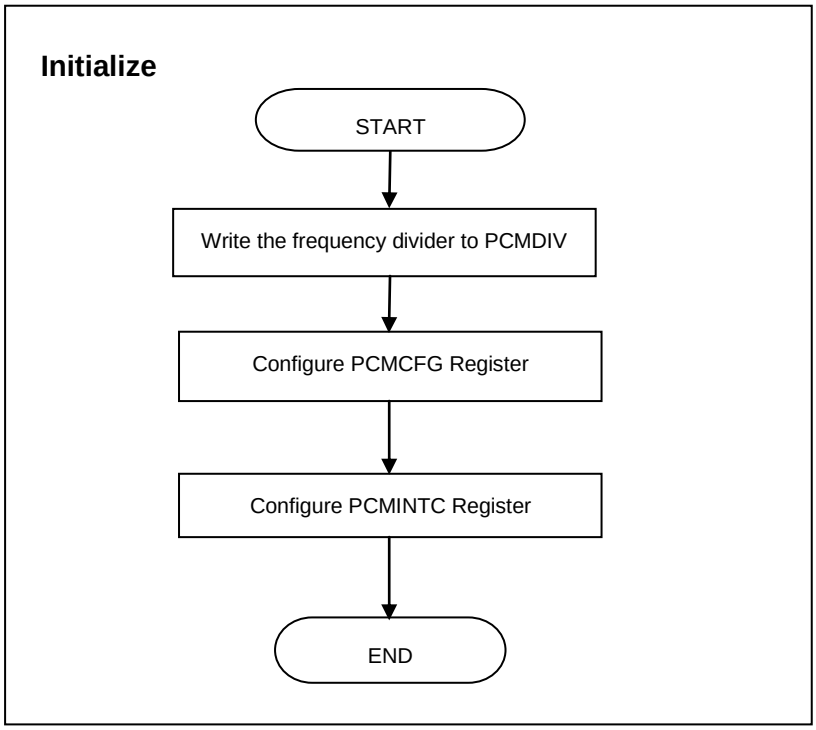


Figure 7-5 Multi-Slot Frame SYN Timing (Shown with two Slots and 8bit Sample)

7.6 PCM Operation

7.6.1 PCM Initialization

At power-on or other hardware reset (WDT and etc), PCM is disabled. Software must initiate PCM after power-on or reset.



For further details, see the corresponding register description sections.

7.6.2 Audio Replay

Outgoing audio sample data is written to PCM transmit FIFO from processor via store instruction or from memory via DMA. PCM then takes the data from the FIFO, serializes it, and sends it over the serial wire PCMDOUT to an external DEVICE.

The audio transmission is enabled automatically when the PCM is enabled by set PCMCTL.PCMEN. And PCMCTL.ERPL must be 1. If PCMCTL.ERPL is 0, value of zero is sent to external DEVICE even if there are samples in transmit FIFO. At least one audio sample data in the transmit FIFO. If the transmit FIFO is empty, value of zero or last sample depends on AICFR.LSMP, is send to external DEVICE even if PCMCTL.ERPL is 1.

Here is the audio replay flow:

1 Configure the external DEVICE as needed.
2 Initialize PCM and configure the register.
3 Write 1 to PCMCTL.PCMEN and PCMCTL.CLKEN.
4 Fill sample data to the transmit FIFO. Repeat this till finish all sample data. In this

procedure, please control the FIFO to make sure no FIFO under-run and other errors happen. When the transmit FIFO under-run, noise or pause may be heard in the audio replay, PCMINTS.TUR is 1, and if PCMINTC.ETUR is 1, PCM issues an interrupt. Please reference to 7.6.4 for detail description on FIFO.

5 Write 1 to PCMCTL.ERPL. It is suggested that at least a frame of PCM data is pre-filled in the transmit FIFO to prevent FIFO under-run flag (PCMINTS.TUR).

6 Waiting for PCMINTS.TFL change to 0. So that all samples in the transmit FIFO has been replayed, then we can have a clean start and write 0 to PCMCTL.ERPL.

7.6.3 Audio Record

Incoming audio sample data is received from PCMDIN serially and converted to parallel word and stored in PCM receive FIFO. Then the data can be taken from the FIFO to processor via load instruction or to memory via DMA.

The audio recording is enabled automatically when the PCM is enabled by set PCMCTL.PCMEN, And PCMCTL.EREC must be 1. If PCMCTL.EREC is 0, the received data is discarded even if there are rooms in the receive FIFO. At least one room left in the receive FIFO. If the receive FIFO is full, the received data is discarded even if PCMCTL.EREC is 1.

Here is the audio record flow:

1 Configure the external DEVICE as needed.
   a Initialize PCM and configure the register.
   b Write 1 to PCMCTL.PCMEN and PCMCTL.CLKEN.
2 Write 1 to PCMCTL.EREC. Make sure there are rooms available in the receive FIFO before set PCMCTL.EREC. Usually, it should empty the receive FIFO by fetch data from it before set PCMCTL.EREC.
3 Take sample data form the receive FIFO. Repeat this till the audio finished. In this procedure, please control the FIFO to make sure no FIFO over-run and other errors happen. When the receive FIFO over-run, same recorded audio samples will be lost, PCMINTS.ROR is 1, and if PCMINTC.EROR is 1, PCM issues an interrupt. Please reference to 7.6.4 for detail description on FIFO.
4 Write 0 to AICCR.EREC.
5 Take sample data from the receive FIFO until PCMINTS.RFL change to 0. So that all samples in the receive FIFO has been taken away, then we can have a clean start up next time. When the receive FIFO is empty, read from it returns zero.

7.6.4 FIFOs operation

PCM has two FIFOs, one for transmitting and one for receiving. The FIFOs are in 16 bits width and 16 entries depth, one entry for keep one sample regardless of the sample size. PCMDP.DATA provides the access point for processor/DMA to write to transmit FIFO and read from receive FIFO. One time access to PCMDP.DATA process one sample. The sample data should be put in LSB

(Least Significant Bit) in memory or processor registers. For transmitting, bits exceed sample are discarded. For receiving, these bits are set to 0. Figure 7-6 illustrates the FIFOs access.

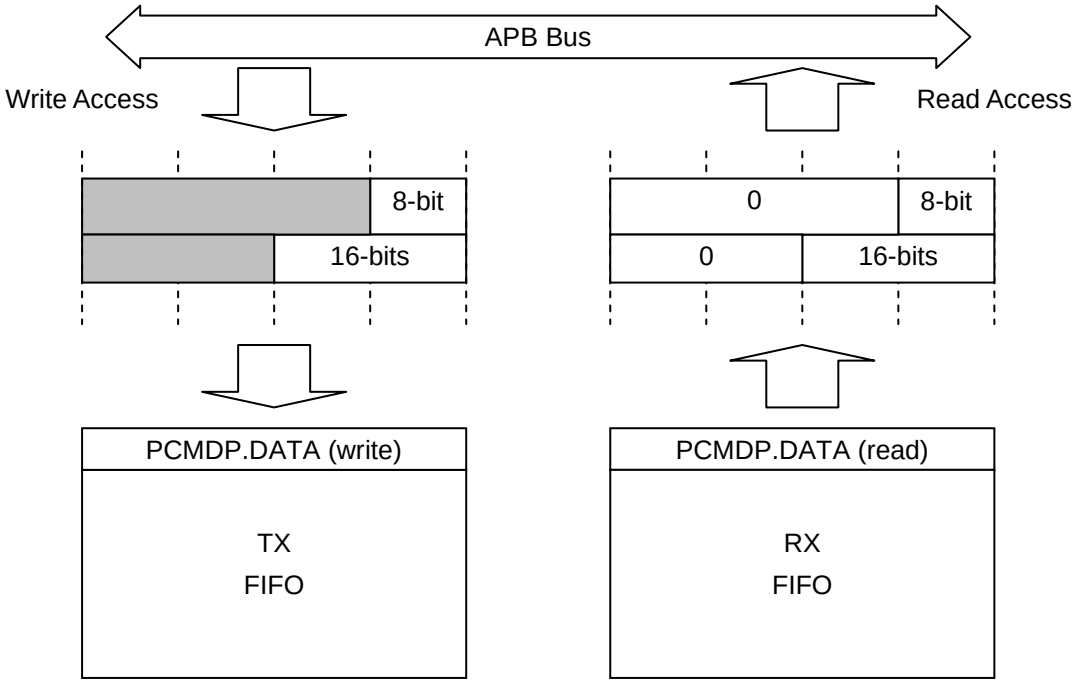


Figure 7-6 Transmitting/Receiving FIFO access via APB Bus

The software and bus initiator must guarantee the right sample placement at the bus.
In case of DMA bus initiator, One 16 bits sample occupies one 16-bits half word in memory, so 16-bits width DMA must be used. One 8-bits sample occupies one byte in memory, and use 8-bits width DMA.

7.6.5 Data Flow Control

There are three approaches provided to control/synchronize the incoming/outgoing data flow.

7.6.5.1 Polling and Processor Access

PCMINTS.RFL and PCMINTS.TFL reflect how many samples exist in receiving and transmitting FIFOs. Through read these register fields, processor can detect when there are samples in receiving FIFO and then load them from the RxFIFO, and when there are rooms in transmitting FIFO and then store samples to the TxFIFO.

Polling approach is in very low efficiency and is not recommended.

7.6.5.2 Interrupt and Processor Access

Set proper values to PCMCFG.TFTH and PCMCFG.RFTH, the FIFO interrupts trig thresholds. Set

PCMINTC.ETFS and/or PCMINTC.ERFS to 1 to enable transmitting and/or receiving FIFO level trigger interrupts. When the interrupt found, it means there are rooms or samples in the TX or RX FIFO, and processor can store or load samples to or from the FIFO.

Interrupt approach is more efficient than polling approach.

7.6.5.3 DMA Access

To enable DMA operation, set PCMCTL.ERDMA and PCMCTL.ETDMA to 1 for transmit and receive respectively. It also needs to allocate two channels in DMA controller for data transmitting and receiving respectively. Please reference to DMAC spec for the details.

The PCMCFG.TFTH and PCMCFG.RFTH are used to set the transmitting and receiving FIFO level thresholds, which determine the issuing of DMA request to DMA controller. To respond the request, DMAC initiator and controls the data movement between memory and TX/RX FIFO.

7.6.6 PCM Serial Clocks and Sampling Frequencies

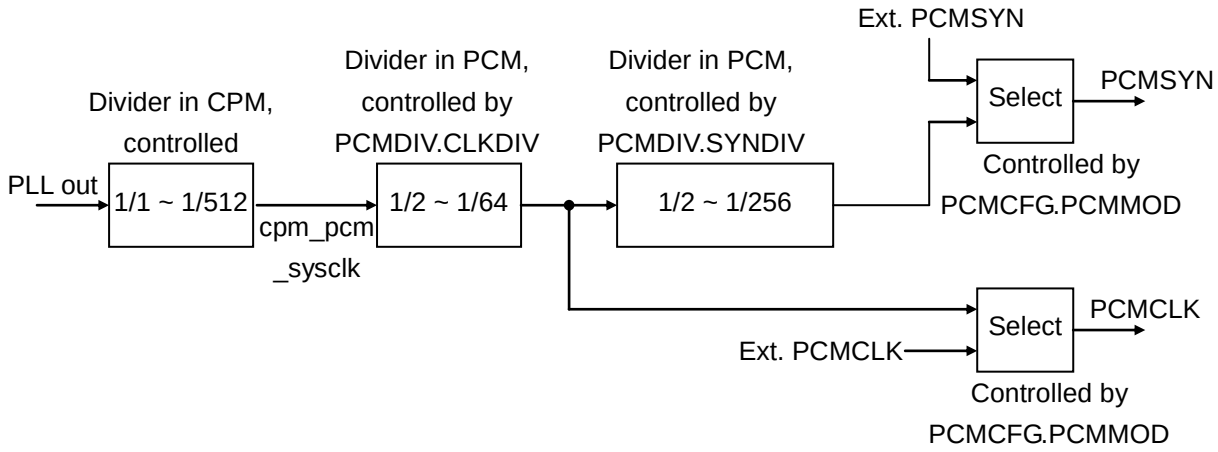


Figure 7-7 PCMCLK and PCMSYN generation scheme

7.6.7 Interrupts

The following status bits, if enabled, interrupt the processor:

- Receive FIFO Service (PCMINTS.RFS). It's also DMA Request.
- Transmit FIFO Service (PCMINTS.TFS). It's also DMA Request.
- Transmit Under-Run (PCMINTS.TUR).
- Receive Over-Run (PCMINTS.ROR).

For further details, see the corresponding register description sections.

8 Internal CODEC

8.1 Overview

This chapter describes the embedded audio CODEC in the processor and related software **interfaces.**

This embedded CODEC is an I2S audio CODEC. AIC module is an interface to this CODEC for audio data replaying and recording. Several memory mapped registers **in AIC** are used to access this embedded CODEC, **and the CODEC's internal control and configure registers can be accessed by write/read these memory mapped registers using 24 MHz clock.**

8.2 Features

The following are internal CODEC features:

- 24 bits ADC and DAC(digital output)
- PWM line out and can load down to 16 Ohm
- Sample rate supported: 8k, 11.025k, 12k, 16k, 22.05k, 24k, 32k, 44.1k, 48k, 88.2k, 96k, 176.4k, and 192k
- Mono line input
- DAC(digital output and converter to analog by external circuit): SNR: 95dB A-Weighted, THD: -80dB @FS-1dB
- Line input to ADC path: SNR: 90dB A-Weighted, THD: -80dB @FS-1dB
- Separate power-down modes for ADC and DAC path with several shutdown modes
- Reduction of audible glitches systems: Soft Mute mode
- Embedded low noise Linear Regulator
- 1 MIC in path or 1 line in path Maximum (Total 1 analog input)

TBD = parameter or document section to be defined later on
TBC = parameter or document section subject to change
TO BE COMPLETED = section to be filled or subject to change

8.2.1 Signal Descriptions

CODEC has **maximum** 5 analog signal IO pins and 4 power pins on chip. They are listed and described in the following table.

Table 8-1 CODEC signal IO pin description

Pin Names	IO	Pin Description	Power
CODEC_AVDD	S	Analog positive power supply	-
CODEC_AVSS	S	Analog negative power supply 2 Ohms max	-
CODEC_VREFP	AI	Analog negative power supply for ADC part	-
VCAP	AO	Decoupling cap for internal biasing voltage for core part	CODEC_AVDD
AIP	AI	Left channel single-ended or positive analog input	CODEC_AVDD
AIN	AI	Left channel negative analog input 1.Must be left floating in single-ended configuration	CODEC_AVDD
MICBIAS	AO	Electric microphone biasing voltage	CODEC_AVDD
VDDIO_CODEC	S	PWM digital line out IO positive power supply	-
VSSIO_CODEC	S	PWM digital line out IO negative power supply	-
CODEC_PWMLP	DO	PWM digital line out positive left channel	VDDIO_CODEC
CODEC_PWMLN	DO	PWM digital line out negative left channel	VDDIO_CODEC
CODEC_PWMRP	DO	PWM digital line out positive right channel	VDDIO_CODEC
CODEC_PWMRN	DO	PWM digital line out negative right channel	VDDIO_CODEC

NOTES:

1 CODEC_AVDD = 3.3v (typ).

2 Internal signal VREFN is connected to CODEC_AVSS, Internal power supply CODEC_AVSS is connected to CODEC_AVSS.

3 Please refer to data sheet of the chip for details.

4 DMIC_IN is ‘GPIO : G3’ , MIC_CLK is ‘GPIO : R5’. Please refer to GPIO specification for these pins operating.

5 nLR: Internal low noise linear regulator.

8.2.2 Block Diagram

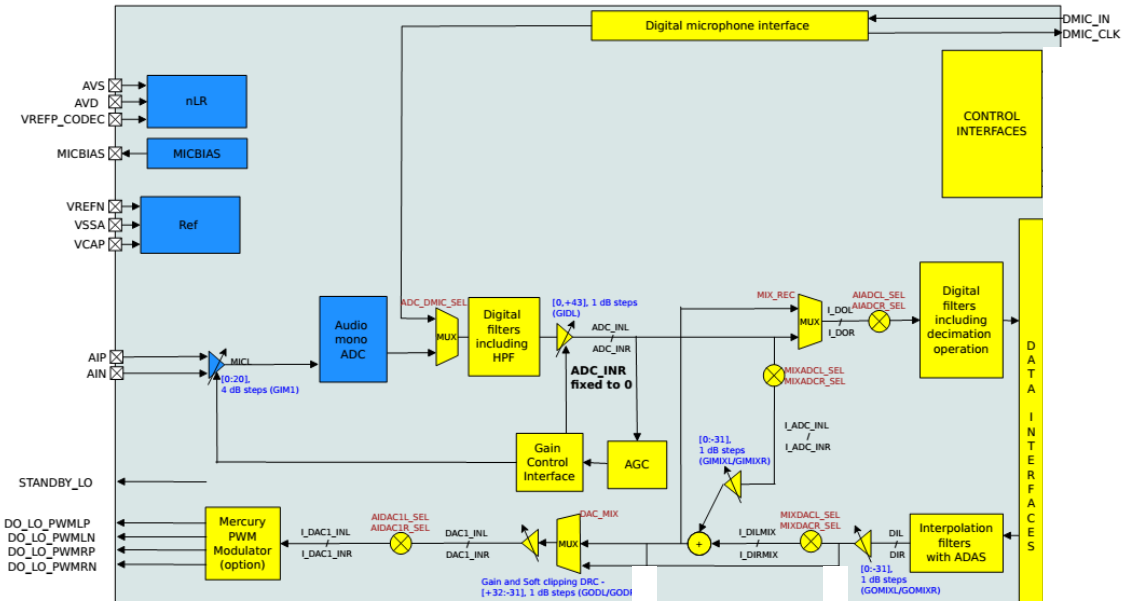


Figure 8-1 CODEC block diagram

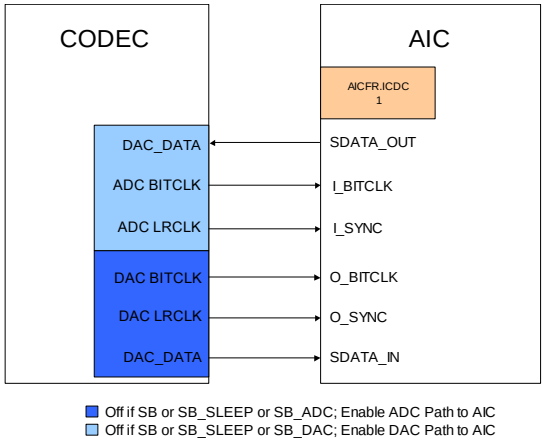
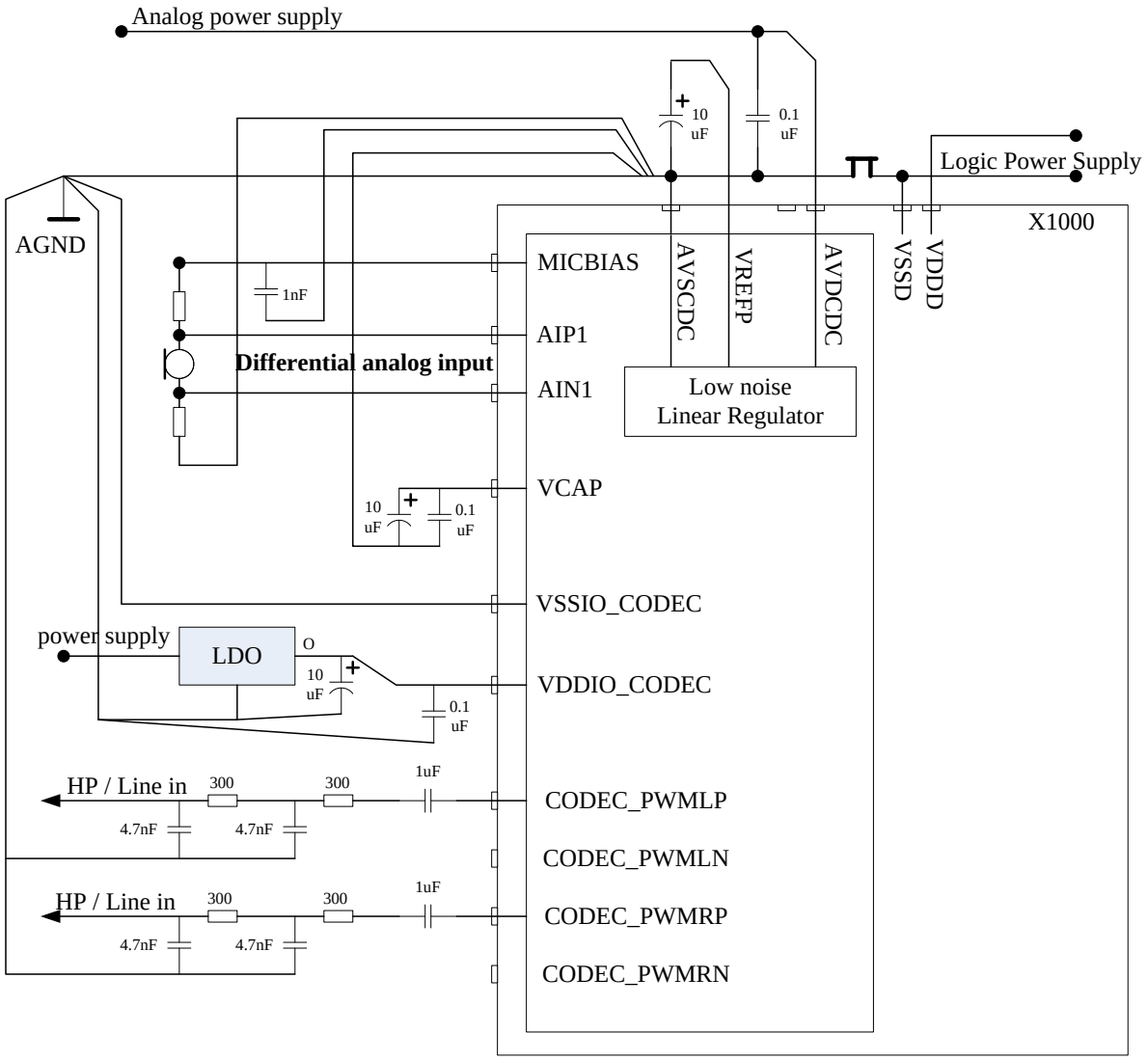


Figure 8-2 Internal CODEC works with AIC

8.2.3 Application schematic



Note: 1. The single-ended/differential input port AIP1/AIN1 can be configure to microphone input or line input by software.

2. VCAP/AVDCDC each of them requires connecting decoupling capacitors (0.1uF) between the pads VCAP/AVDCDC and CODEC_AVSS. This ceramic capacitor has to be kept as close as possible to IC package (closer than 0.2 inch)

8.3 Mapped Register Descriptions

The internal CODEC software interface includes 2 registers. They are mapped **to** IO memory address space of AIC module so that program can access them to control the operations of the CODEC.

Table 8-2 Internal CODEC Mapped Registers Description (AIC Registers)

Name	Description	RW	Reset value	Address	Size
RGADW	Address, data in and write command for accessing to internal registers of internal embedded CODEC.	RW	0x00000000	0x100200A4	32
RGDATA	The data read out and interrupt reque-st status of Internal registers data in the internal embedded CODEC.	R	0x00000000	0x100200A8	32

NOTES:

1 All these registers are AIC Registers, because they are mapped **to** AIC IO memory address.

2 RGADW contains data, address and write command to the internal registers of the internal CODEC.

3 RGDATA returns the internal register value of the internal CODEC and interrupt request status.

8.3.1 CODEC internal register access control (RGADW)

RGADW contains address, data and write command to the internal registers of the internal embedded CODEC.

	RGADW																													0x100200A4
Bit	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
	ICRST	Reserved														RGWR	Reserved	RGADDR							RGDIN							
RST	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

Bits	Name	Description	RW
31	ICRST	If set ‘1’ to this bit internal codec will reset and then clear to ‘0’ to end reset.	RW
30:17	Reserved	Writing has no effect, read as zero.	R
16	RGWR	Writing 1 to this bit means issuing a write command to CODEC’s internal register. This bit keeps 1 by AIC before the current write process is finished. A register read or a new register write cannot be issued before the previous write process is finished. In another word, RGADW should not be written before RGADW.RGWR becomes 0. A write process takes up to the sum of 0.17us and 1 PCLK cycle. Writing 0 to this bit is ignored.	RW
15	Reserved	Writing has no effect, read as zero.	R
14:8	RGADDR	When it issues a writing command to CODEC’s internal register, i.e.RGWR=1, this field specifies the corresponding CODEC register’s address. In addition, this field also decides the address of the CODEC	RW

		register's data out (The data is mapped to RGDATA. RGDOUT) at any time.	
7:0	RGDIN	When it issues a write command to CODEC's internal register, i.e. RGWR=1, this field contains the data to be written to the CODEC register.	RW

NOTES:

1 **It is strongly suggested verifying the data (using read RGDATA below) after writing the data to internal register of CODEC. When verifying RGDATA. RGDOUT right (RGDATA. RGDOUT is the return data from internal CODEC register of the corresponding address), the writing process is finished.**

2 Please notice that AIC needs MCLK (**please refer** to AIC spec), when write new value to or read from CODEC internal registers.

8.3.2 CODEC internal register data output (RGDATA)

RGDATA returns the register value of the internal embedded CODEC and interrupt request status.

	RGDATA																														0x100200A8
Bit	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
	Reserve																							IRQ	RGDOUT							
RST	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

Bits	Name	Description	RW
31:9	Reserved	Writing has no effect, read as zero.	R
8	IRQ	This field returns the internal embedded CODEC's interrupt request. IRQ: 0 — No CODEC's interrupt request found. IRQ: 1 — CODEC's interrupt request is pending.	R
7:0	RGDOUT	This field returns the value of the register in internal embedded CODEC. (RGADW.RGADDR field specifies the CODEC register's address)	R

NOTE: AIC needs MCLK (**please refer** to AIC spec), when write new value to or read from CODEC internal registers.

8.4 Operation

The internal embedded CODEC is controlled **through** its internal registers. These registers **data** can be accessed through memory-mapped registers, RGADW and RGDATA. AIC's BITCLK and

SYNC are **from** the CODEC and is controlled by CKCFG.SELAD register. The audio data transferring, i.e. audio replaying and recording, is done by AIC. AIC still **plays** the role of I2S controller. We will refer to many AIC operations and registers in the following audio operation descriptions, please reference to AIC spec for the details.

This is a guide for software.

8.4.1 Access to internal registers of the embedded CODEC

The embedded CODEC is controlled through its internal registers. RGADW and RGDATA are used to write to and read from these registers. Here are some examples.

Example 1. Write to a CODEC internal register.
Step 1: RGADW.RGWR == 0.
Step 2: If not, go to step 1.
Step 3: Write to RGADW and make it.
RGADW.RGDIN = <data to be written to the register>.
RGADW.RGADDR = <the register's address >.
Step 4: Write to RGADW to commit the writing operation.
RGADW.RGWR = 1.

Example 2. Read from a CODEC internal register.
Step 1: RGADW.RGWR == 0.
Step 2: If not, go to step 1.
Step 3: write to RGADW and make it.
RGADW.RGWR = 0.
RGADW.RGDIN = <don't care>.
RGADW.RGADDR = <the register's address>.
Step 4: read RGDATA.DOUT, which returns the register's content.

8.4.2 CODEC controlling and typical operations

This section is **about** some typical operations. We assume the power supply of CODEC is on, and CODEC is in STANDBY mode.

Before **performing** any of these operations, make sure AIC is configured properly as list below:

1 Make AIC to use internal CODEC mode:
AICFR.ICDC = 1; Use internal CODEC.
AICFR.AUSEL = 1; Use I2S mode.
AICFR.BCKD = 0; CODEC input BIT_CLK to AIC.
AICFR.SYNCD = 0; CODEC input SYNC to AIC.
I2SCR.AMSL = 1; Use I2S operation mode.
I2SCR.ESCLK = 1; Open MCLK to internal CODEC. (if using PLL Clock)

2 Make sure AICCR.FLUSH = 0; AICFR.RST = 0; AICCR.ENLBF = 0.
3 Clear AICSR.ROR, AICSR.TUR, AICSR.RFS, AICSR.TFS to 0.
4 Set proper value to AICCR.M2S; AICCR.ENDSW; AICCR.ASVTSU.
5 Set AICFR.ENB to 1; Open AIC.

When using DMA mode, configure AICFR.RFTH, AICCR.RDMS or AICFR.TFTH, AICCR.TDMS.
Configure TX-FIFO and interrupt means setting proper value to AICFR.TFTH, clear AICCR.ETFS to 0, and clear AICCR.ETUR to 0.
Configure RX-FIFO and interrupt means setting proper value to AICFR.RFTH, clear AICCR.ERFS to 0 and clear AICCR.EROR to 0.
When configure interrupt, software must handle all the interrupts. So all interrupts are recommended disabled as shown above.
CODEC shares the interrupt with AIC module.

The register or register bit of CODEC will use the same form as the Mapped registers, but software should use the method in the section "Mapped Register Descriptions" to access this registers.

More details are listed in the CODEC guide.

8.4.3 Power saving

There are many power modes in CODEC. In every working mode, it should close **unused** stages (parts) of CODEC for saving power.

The power diagram is shown in "CODEC Power Diagram"; please refer to "CODEC Operating modes".

8.4.4 Pop noise and the reduction of it

Please refer to "Ramping system note" and "Anti-pop operation sequences" for details.

8.4.4.1 Reference open step

1 Init play.

Step 0: Open DMA and two AIC modules Clocks in CPM.CLKGR.
Step 1: Configure AIC as slave and using internal CODEC mode.
AICFR.ICDC = 1; Use internal CODEC.
AICFR.AUSEL = 1; Use I2S mode.
AICFR.BCKD = 0; CODEC input BIT_CLK to AIC.
AICFR.SYNCD = 0; CODEC input SYNC to AIC.
I2SCR.AMSL = 1; Use I2S operation mode.
I2SCR.ESCLK = 1; Open MCLK to internal CODEC.

Step 2: Configure DMA as slave mode using internal CODEC.

2 Open.

Step 0: Enable DMA Channel Clock.

Step 1: Configure AIC sample size and sample rate. Configure AIC Output FIFO Threshold.

Step 2: Configure DMA.

Step 3: Configure CODEC.

3 Write.

Step 0: Enable DMA Channel Clock.

Step 1: Configure AIC.

Step 2: Configure DMA.

Step 3: Configure CODEC.

4 Read.

Step 0: Enable DMA Channel Clock.

Step 1: Configure AIC.

Step 2: Configure DMA.

Step 3: Configure CODEC.

5 Close.

6 End.

NOTES:

1 SB_DAC Control the internal OBIT_CLK from CODEC to AIC, First turn it on when write data (replay).

2 SB_ADC Control the internal IBIT_CLK from CODEC to AIC, First turns it on when read data (record).

8.5 Timing parameters

Parameter	Condition	Min.	Typ.	Max.	Unit
Tsbyu	Cext = 10uF/100nF +/-20%		250	600	ms
Tshd_adc	Cext = 10uF/100nF +/-20%		200		ms
Tshd_dac	Cext = 10uF/100nF +/-20%		400	900	ms
Tr, Tf (all inputs)	All modes			5	ns
Tr, Tf (all outputs)	All modes			5	ns

NOTES:

1 Tsbyu is the reference wake-up time after complete power down.

2 Tshd_adc is the ADC wake-up time after sleep mode.

3 Tshd_dac is DAC wake-up time after sleep mode.

8.6 C parameters

Voltages:

CODEC_AVSS is connected to analog ground.

AVDCDC= 3.3V (typ).

power consumption

mode	power consumption	Unit
Sleep mode	1200	uW
Playback stereo audio DAC only (capacitor coupled load configuration)	8.3	mW
Record mic stereo input only (audio ADC)	10.3	mW
Record mic mono input only (audio ADC)	6.3	mW
Bypass path (audio stereo line in to HP)	7.7	mW

Current value is at AVDCDC = 3.3 V.

Chip pin Name	MAX Current across I/O @ AVDCDC = 3.3 V
AVDCDC	< 200 mA in normal working mode
CODEC_AVSS	< 20 mA in normal working mode
AVSHP	< 160 mA in normal working mode
	< 1400 mA in case of short circuit
VCAP	< 2 mA in normal working mode
AIP1,AIN1	< 2 mA in normal working mode
AIP2,AIN2	< 2 mA in normal working mode
AIP3,AIN3	< 2 mA in normal working mode
AIP4,AIN4	< 2 mA in normal working mode
MICBIAS1	< 5 mA in normal working mode
MICBIAS2	< 5 mA in normal working mode
AOHPL	< 80 mA in normal working mode
	< 1200 mA in case of short circuit
AOHPR	< 80 mA in normal working mode
	< 1200 mA in case of short circuit
AOLOP,AOLON	< 1 mA in normal working mode
HPSENSE	< 1 mA in normal working mode

The current in case of short circuit is the max value. This current is only sink or drawn until the short circuit detection system acts.

Please refer to Chip Datasheet for more details.

8.7 CODEC internal Registers

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute
   R - Read only
   W - Write only
   RW - read and write
   RCW - read and write, but clear to 0 by read
   RSW - read and write, but set to 1 by read
   RWC - read and write, clear to 0 by write 1, write 0 has no effect
   RWS - read and write, set to 1 by write 1, write 0 has no effect
   RC - read only, and clear to 0 by read
   RS - read only, and set to 1 by read
   SPEC - special access method, relate to its description
3. Reset Value
   1 - reset to 1
   0 - reset to 0
   ? - value unknown after reset

8.7.1 Registers Memory Map

Register Name	Function	Address	Reset value	Access Size
SR	Status register	0x00	-	8
SR2	Status register 2	0x01	-	8
SIGR	Signature register	0x02		8
SIGR2	Signature register 2	0x03		8
SIGR3	Signature register 3	0x04		8
SIGR5	Signature register 5	0x05		8
SIGR7	Signature register 7	0x06		8
MR	Mode status register	0x07	-	8
AICR_DAC	DAC audio interface control register	0x08	hD3	8
AICR_ADC	ADC audio interface control register	0x09	hD3	8
CR_DMIC	Digital microphone control register	0x0A	h00	8
CR_MIC1	Microphone1 control register	0x0B	h30	8
CR_MIC2	Microphone2 control register	0x0C	h30	8
CR_DAC	DAC control register	0x0D	hB0	8
CR_DAC2	DAC control register	0x0E	hB1	8
CR_ADC	ADC control register	0x0F	hB0	8
CR_MIX	Digital mixer control register	0x10	h00	8
DR_MIX	Digital mixer data register	0x1A	h00	8
MIX_0	Digital mixer control register 0	-	h00	8

MIX_1	Digital mixer control register 1	-	h00	8
MIX_2	Digital mixer control register 2	-	h00	8
MIX_3	Digital mixer control register 3	-	h00	8
MIX_4	Digital mixer control register 4	-	h00	8
CR_VIC	Control register for the CODEC	0x12	h0F	8
CR_CK	Clock control register	0x13	h40	8
FCR_DAC	DAC frequency control register	0x14	h00	8
SFCCR_DAC	DAC sample frequency fine control register	0x15	h00	8
SFFCR_DAC	DAC sample frequency fine control register	0x16	h00	8
FCR_ADC	ADC Frequency control register	0x17	h00	8
CR_TIMER_MSB	MSB of programmable counter	0x18	h00	8
CR_TIMER_LSB	LSB of programmable counter	0x19	h00	8
ICR	Interrupt control register	0x1A	h00	8
IMR	Interrupt mask register	0x1B	hFF	8
IFR	Interrupt flag register	0x1C	-	8
IMR2	Interrupt mask register 2	0x1D	hFF	8
IFR2	Interrupt flag register	0x1E	-	8
GCR_DACL	Left channel DAC gain control register	0x1F	h00	8
GCR_DACR	Right channel DAC gain control register	0x20	h00	8
GCR_DAC2L	Left channel DAC gain control register	0x21	h00	8
GCR_DAC2R	Right channel DAC gain control register	0x22	h00	8
GCR_MIC1	Microphone 1 gain control register	0x23	h00	8
GCR_MIC2	Microphone 2 gain control register	0x24	h00	8
GCR_ADCL	Left ADC gain control register	0x25	h00	8
GCR_ADCR	Right ADC gain control register	0x26	h00	8
GCR_MIXDACL	Left DAC digital mixer gain control register	0x27	h00	8
GCR_MIXDACR	Right DAC digital mixer gain control register	0x28	h00	8
GCR_MIXADCL	Left ADC digital mixer gain control register	0x29	h00	8
GCR_MIXADCR	Right ADC digital mixer gain Control register	0x2A	h00	8
CR_DAC_AGC	DAC soft clipping DRC control register	0x2B	h00	8
DR_DAC_AGC	DAC soft clipping DRC data register	0x2C	h00	8
DAC_AGC_0	DAC soft clipping DRC control register 0	-	h00	8
DAC_AGC_1	DAC soft clipping DRC control register 1	-	h00	8
DAC_AGC_2	DAC soft clipping DRC control register 2	-	h00	8
DAC_AGC_3	DAC soft clipping DRC control register 3	-	h00	8
CR_DAC2_AGC	DAC2 soft clipping DRC control register	0x2D	h00	8
DR_DAC2_AGC	DAC2 soft clipping DRC data register	0x2E	h00	8
DAC2_AGC_0	DAC2 soft clipping DRC control register 0	-	h00	8
DAC2_AGC_1	DAC2 soft clipping DRC control register 1	-	h00	8
DAC2_AGC_2	DAC2 soft clipping DRC control register 2	-	h00	8
DAC2_AGC_3	DAC2 soft clipping DRC control register 3	-	h00	8
CR_ADC_AGC	ADC automatic gain control register	0x2F	h00	8

DR_ADC_AGC	ADC automatic gain control data register	0x30	h00	8
ADC_AGC_0	ADC automatic gain control register 0	-	h34	8
ADC _AGC_1	ADC automatic gain control register 1	-	h07	8
ADC _AGC_2	ADC automatic gain control register 2	-	h44	8
ADC _AGC_3	ADC automatic gain control register 3	-	h1F	8
ADC _AGC_4	ADC automatic gain control register 4	-	h00	8
SR_ADC_AGCDGL	ADC AGC left digital gain	0x31	-	8
SR_ADC_AGCDGR	ADC AGC right digital gain	0x32	-	8
SR_ADC_AGCAGL	ADC AGC left analog gain	0x33	-	8
SR_ADC_AGCAGR	ADC AGC right analog gain	0x34	-	8
CR_TR	Test mode control register	0x35	h00	8
DR_TR	Test mode data register	0x36	h00	8
SR_TR1	Test mode status register	0x37	-	8
SR_TR2	Test mode status register 2	0x38	-	8
SR_TR_SRCDAC	DAC test mode status register	0x39	-	8

8.7.2 Register Description

8.7.2.1 SR: Status Register

Register Name: SR				Register Address: 0x00			
bit7-R-?	bit6-R-?	bit5-R-?	bit4-R-?	bit3-R-?	bit2-R-?	bit1-R-?	bit0-R-?
PON_ACK	IRQ_ACK	Reserved	DAC_LOCKED	Reserved			

Bits	Field	Description
7	PON_ACK	Acknowledge status bit after power on. Read 0: reset value 1: codec is ready to operate
6	IRQ_ACK	Acknowledge status bit after IRQ sending. Read 0: reset value 1: codec has requested an interrupt (IRQ signal activated)
5	Reserved	Writing has no effect, read as zero.
4	DAC_LOCKED	Acknowledge status bit after Sample Rate Converter system is locked Read 0: SRC is not locked. Data from the audio interface are automatically muted. 1: The SRC is locked and operating normally.
3:0	Reserved	Writing has no effect, read as zero.

8.7.2.2 SR2: Status Register 2

Register Name: SR2				Register Address: 0x01			
bit7-R-?	bit6-R-?	bit5-R-?	bit4-R-?	bit3-R-?	bit2-R-?	bit1-R-?	bit0-R-?
Reserved			DAC_UNKNOWN_FS	Reserved			

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4	DAC_UNKOWN_FS	Read 0: Measured mean frequency of incoming audio data is within supported values 1: Measured mean frequency of incoming audio data is not within supported values
3:0	Reserved	Writing has no effect, read as zero.

8.7.2.3 SIGR: Signature Register

Register Name: SIGR				Register Address: 0x02			
bit7-R-?	bit6-R-?	bit5-R-?	bit4-R-?	bit3-R-?	bit2-R-?	bit1-R-?	bit0-R-?
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect, read as zero.

8.7.2.4 SIGR2: Signature Register 2

Register Name: SIGR2				Register Address: 0x03			
bit7-R-?	bit6-R-?	bit5-R-?	bit4-R-?	bit3-R-?	bit2-R-?	bit1-R-?	bit0-R-?
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect, read as zero.

8.7.2.5 SIGR3: Signature Register 3

Register Name: SIGR3				Register Address: 0x04			
bit7-R-?	bit6-R-?	bit5-R-?	bit4-R-?	bit3-R-?	bit2-R-?	bit1-R-?	bit0-R-?
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect, read as zero.

8.7.2.6 SIGR5: Signature Register 5

Register Name: SIGR5				Register Address: 0x05			
bit7-R-?	bit6-R-?	bit5-R-?	bit4-R-?	bit3-R-?	bit2-R-?	bit1-R-?	bit0-R-?
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect, read as zero.

8.7.2.7 SIGR7: Signature Register 7

Register Name: SIGR7				Register Address: 0x06			
bit7-R-?	bit6-R-?	bit5-R-?	bit4-R-?	bit3-R-?	bit2-R-?	bit1-R-?	bit0-R-?
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect, read as zero.

8.7.2.8 MR: Status Register

Register Name: MR				Register Address: 0x07			
bit7-R-?	bit6-R-?	bit5-R-?	bit4-R-?	bit3-R-?	bit2-R-?	bit1-R-?	bit0-R-?
Reserved			ADC_MUTE		Reserved	DAC_MUTE	

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:3	ADC_MUTE	Read 00 : ADC not muted 01 : ADC being muted 10 : ADC leaving mute mode 11 : ADC in mute mode
2	Reserved	Writing has no effect, read as zero.
1:0	DAC_MUTE	Read 00 : DAC not muted 01 : DAC being muted 10 : DAC leaving mute mode 11 : DAC in mute mode

8.7.2.9 AICR_DAC: Audio Interface Control Register

Register Name: AICR_DAC				Register Address: 0x08			
bit7-RW-1	bit6-RW-1	bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-1	bit0-RW-1
DAC_ADWL		DAC_SLAVE	SB_AICR_DAC	Reserved		DAC_AUDIOIF	

Bits	Field	Description
7:6	DAC_ADWL	Audio Data Word Length for DAC path. Read / Write 00: 16-bit word length data 01: 18-bit word length data 10: 20-bit word length data 11: 24-bit word length data
5	DAC_SLAVE	Read/Write: 0 = Audio interface in master mode 1 = Audio interface in slave mode
4	SB_AICR_DAC	Read/Write: 0 = DAC audio interface active 1 = DAC audio interface in power-down mode
3:2	Reserved	Writing has no effect, read as zero.
1:0	DAC_AUDIOIF	Audio Interface mode selection for DAC Read/Write: 00 = Parallel interface 01 = Left-justified interface 10 = DSP interface 11= I2S interface

NOTES:

1 DAC_AUDIOIF should be configured to 11

8.7.2.10 AICR_ADC: Audio Interface Control Register

Register Name: AICR_ADC				Register Address: 0x09			
bit7-RW-1	bit6-RW-1	bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-1	bit0-RW-1
ADC_ADWL		Reserved	SB_AICR_ADC	Reserved		ADC_AUDIOIF	

Bits	Field	Description
7:6	ADC_ADWL	Audio Data Word Length for ADC path. Read / Write 00: 16-bit word length data 01: 18-bit word length data 10: 20-bit word length data

		11: 24-bit word length data
5	Reserved	Writing has no effect, read as zero.
4	SB_AICR_ADC	Read/Write: 0 = ADC audio interface active 1 = ADC audio interface in power-down mode
3:2	Reserved	Writing has no effect, read as zero.
1:0	ADC_AUDIOIF	Audio Interface mode selection for ADC Read/Write: 00 = Parallel interface 01 = I2S interface 10 = DSP interface 11 = I2S interface

NOTES:

1 ADC_AUDIOIF should be configured to 11

8.7.2.11 CR_DMIC: Digital Microphone Control Register

Register Name: CR_DMIC				Register Address: 0x0A			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
DMIC_CLKON	Reserved						

Bits	Field	Description
7	DMIC_CLKON	Digital microphone clock (DMIC_CLK) enable Read/Write 0= clock off 1= clock on, clock frequency varies with DMIC_RATE and MCLK
6:0	Reserved	Writing has no effect, read as zero.

8.7.2.12 CR_MIC1: Control Register for microphone1 inputs

Register Name: CR_MIC1				Register Address: 0x0B			
bit7-RW-1	bit6-RW-0	bit5-RW-1	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved	MICIDFF1	SB_MICBIAS1	SB_MIC1	MICBIAS1_V	Reserved		

Bits	Field	Description
7	Reserved	Writing has no effect, read as zero.
6	MICIDFF1	Microphone 1 input mode selection Read/Write 0= single-ended input

		1= differential input
5	SB_MICBIAS1	Microphone 1 biasing buffer power-down Read/Write 0 = active 1 = power-down
4	SB_MIC1	Microphone 1 power-down Read/Write 0 = active 1 = power-down
3	MICBIAS1_V	Micbias level detection 0：MICBIAS.= 2.08V 1：MICBIAS.= 1.66V
2：	Reserved	Writing has no effect, read as zero.

8.7.2.13 CR_MIC2: Control Register for microphone2 inputs

Register Name: CR_MIC2				Register Address: 0x0C			
bit7-RW-0	bit6-RW-0	bit5-RW-1	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved							

Bits	Field	Description
7:6	Reserved	Writing has no effect, read as zero.
5:4	Reserved	Writing has no effect, read as one..
3:0	Reserved	Writing has no effect, read as zero.

8.7.2.14 CR_DAC: Control Register for DAC

Register Name: CR_DAC				Register Address: 0x0D			
bit7-RW-1	bit6-RW-0	bit5-RW-1	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
DAC_SOFT_MUTE	Reserved		SB_DAC	Reserved			DAC_ZERO_N

Bits	Field	Description
7	DAC_SOFT_MUTE	DAC soft mute mode. Read/Write 0: mute inactive, digital input signal transmitted to the DAC 1: puts the DAC in soft mute mode
6	Reserved	Writing has no effect, read as zero.
5	Reserved	Writing has no effect, read as one..

4	SB_DAC	DAC power-down mode. Read/Write 0: active 1: power-down
3:1	Reserved	Writing has no effect, read as zero.
0	DAC_ZERO_N	Disables DO_LO_PMM*N output 0:Ouputs are activated 1:Ououts are stuck to VSSD

8.7.2.15 CR_DAC2: Control Register for DAC2

Register Name: CR_DAC2				Register Address: 0x0E			
bit7-RW-1	bit6-RW-0	bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved		DAC2_LEFT_ONLY	SB_DAC2	DAC2_ZERO_N	DAC2_DIT		

Bits	Field	Description
7	Reserved	Writing has no effect, read as one.
6	Reserved	Writing has no effect, read as zero.
5	DAC2_LEFT_ONLY	Deactivation of DAC2 Titanium Modulator right channel 0: audio DACL and audio DACR active 1: power-down
4	SB_DAC2	DAC power-down mode. Read/Write 0: active 1: power-down
3	DAC2_ZERO_N	Disables DO_LO_PMM*N output 0:Ouputs are activated 1:Ououts are stuck to VSSD
2:0	DAC2_DIT	Selection of spread spectrum(Lower EMI)for Titanium Modulator 000: Minimum value: 1 111: Maximum value: 6

8.7.2.16 CR_ADC: Control Register for ADC

Register Name: CR_ADC				Register Address: 0x0F			
bit7-RW-1	bit6-RW-0	bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
ADC_SOFT_MUTE	ADC_DMIC_SEL	Reserved	SB_ADC	Reserved			

Bits	Field	Description
7	ADC_SOFT_MUTE	ADC soft mute mode Read/Write 0 = mute inactive 1 = puts the ADC in soft mute mode
6	ADC_DMIC_SEL	digital filter input selection Read/Write 0= ADC 1= Digital microphone
5	Reserved	Writing has no effect, read as zero.
4	SB_ADC	ADC power down mode. Read/Write 0: active 1: power-down
3:0	Reserved	Writing has no effect, read as zero.

NOTE: When ADC right channel is deactivated, right channel output is set to 0

8.7.2.17 CR_MIX: Control Register for digital mixer

Register Name: CR_MIX				Register Address: 0x10			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
MIX_EN	MIX_LOAD	MIX_ADD					

Bits	Field	Description
7	MIX_EN	digital mixer enable Read/Write 0: digital mixer off 1: digital mixer enabled
6	MIX_LOAD	Read/Write Refer to DR_MIX for MIX_LOAD description
5:0	MIX_ADD	Digital mixer control registers address Read/Write

MIX_ADD	Corresponding control register
000000	MIX_0
000001	MIX_1
000010	MIX_2
000011	MIX_3

8.7.2.18 DR_MIX: Data register for digital mixer

Register Name: CR_MIX				Register Address: 0x11			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
MIX_DATA							

Bits	Field	Description
7:0	MIX_DATA	● Instructions for writing into a register The data must firstly be written into the DR_MIX register. The register address must then be written in CR_MIX (MIX_ADD[5:0]). The data is written into the register when MIX_LOAD is set to '1'. ● Instructions for reading the values in a register The address of the register to be read is written in CR_MIX (MIX_ADD[5:0]). The content of the corresponding register appears in DR_MIX when MIX_LOAD is set to '0'.

MIX_0: Digital mixer control register 0

Register Name: CR_MIX				Indirect register Address: 0x00			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
AIDACL_SEL		AIDACR_SEL		Reserved			DAC_MIX

Bits	Field	Description
7:4	AIDACL_SEL	DAC input selection Read/Write 00: cross inputs 01: normal inputs 10: mixed inputs 11: 0 inputs
3:1	Reserved	Writing has no effect, read as zero.
0	DAC_MIX	Mixer mode on DAC Path Read/Write 0: Playback DAC only 1: Playback DAC + ADC

Note: AIDACX_SEL should be configured to 01 in normal mode.

MIX_1: Digital mixer control register 1

Register Name: CR_MIX				Indirect register Address: 0x01			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
MIXDACL_SEL		MIXDACR_SEL		Reserved			

Bits	Field	Description
7:4	MIXDACX_SEL	Mixer input selection on DAC path Read/Write 00: normal inputs 01: cross inputs 10: mixed inputs 11: 0 inputs
3:0	Reserved	Writing has no effect, read as zero.

MIX_2: Digital mixer control register 2

Register Name: CR_MIX				Indirect register Address: 0x02			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
AIADCL_SEL		AIADCR_SEL		Reserved			MIX_REC

Bits	Field	Description
7:4	AIADCX_SEL	Read/Write 00: normal inputs 01: cross inputs 10: mixed inputs 11: 0 inputs
3:1	Reserved	Writing has no effect, read as zero.
0	MIX_REC	Mixer mode on ADC Path Read/Write 0: Record input only 1: Record input + DAC

MIX_3: Digital mixer control register 3

Register Name: CR_MIX				Indirect register Address: 0x03			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
MIXADCL_SEL		MIXADCR_SEL		Reserved			

Bits	Field	Description
7:6	MIXADCX_SEL	Read/Write 00: cross inputs 01: normal inputs 10: mixed inputs 11: 0 inputs
3:0	Reserved	Writing has no effect, read as zero.

Note: MIXADCX_SEL should be configured to 01 in normal mode.

MIX_4: Digital mixer control register 4

Register Name: CR_MIX				Indirect register Address: 0x04			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
AIDAC2L_SEL		AIDAC2R_SEL		Reserved			

Bits	Field	Description
7:4	MIXADC2X_SEL	Read/Write 00: cross inputs 01: normal inputs 10: mixed inputs 11: 0 inputs
3:0	Reserved	Writing has no effect, read as zero.

8.7.2.19 CR_VIC: Control Register for the CODEC

Register Name: CR_VIC				Register Address: 0x12			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-1	bit2-RW-1	bit1-RW-1	bit0-RW-1
Reserved						SB_SLEEP	SB

Bits	Field	Description
7:4	Reserved	Writing has no effect, read as zero.
3:2	Reserved	Writing has no effect, read as one.
1	SB_SLEEP	sleep mode. Read/Write 0: normal mode (active) 1: sleep mode
0	SB	complete power-down mode. Read/Write 0: normal mode (active) 1: complete power-down

8.7.2.20 CR_CK: Clock Control Register

Register Name: CR_CK				Register Address: 0x13			
bit7-RW-0	bit6-RW-1	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved	MCLK_DIV	Reserved	SHUTDOWN_ CLOCK	CRYSTAL			

Bits	Field	Description
7	Reserved	Writing has no effect, read as zero.

6	MCLK_DIV	0: Division of MCLK by 1(Reserved for Dolphin usage) 1: Division of MCLK by 2
5	Reserved	Writing has no effect, read as zero.
4	SHUTDOWN_ CLOCK	Read/Write 0= normal mode (MCLK active) 1= MCLK is turned off (stuck to '0')
3:0	CRYSTAL	Selection of the MCLK frequency Read/Write The sampling frequency value is given in the CRYSTAL[3:0] table

CRYSTAL[3:0]	Master Clock Frequency
0000	12 MHz
0001	Reserved for further use
0010	13 MHz
….	Reserved for further use
1111	Reserved for further use

NOTE: CRYSTAL should be configured to 0x0 for setting the internal 12MHz master clock MCLK (default).

8.7.2.21 FCR_DAC: DAC Frequency Control Register

Register Name: FCR_DAC				Register Address: 0x14			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved				DAC_FREQ			

Bits	Field	Description
7:4	Reserved	Writing has no effect, read as zero.
3:0	DAC_FREQ	Selection of the DAC sampling rate (Fs). Read/Write The sampling frequency value is given in the FREQ table.

NOTE: Please refer to section Sample frequency: FREQ.

8.7.2.22 SFCCR_DAC: DAC Sample frequency fine control register

Register Name: SFCCR_DAC				Register Address: 0x15			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
DACFREQ_ VALID	DAC_FREQ_ADJ[14:8]						

Bits	Field	Description
7	DACFREQ_VALID	Enables the sampling frequency adjustment (DAC_FREQ_ADJ[14:0]) in master mode. Bits DAC_FREQ_ADJ[14:0] are split in both registers SFCCR_DAC and SFFCR_DAC.
6:0	DAC_FREQ_ADJ [14:8]	Frequency Tuning Control Word. DAC_FREQ_ADJ allows a fine tuning of DAC audio data sampling frequency in master mode.

8.7.2.23 SFFCR_DAC: SRC frequency adjustment register 2

Register Name: SFFCR_DAC				Register Address: 0x16			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
DAC_FREQ_ADJ[7:0]							

Bits	Field	Description
7:0	DAC_FREQ_ADJ [7:0]	Frequency Tuning Control Word. DAC_FREQ_ADJ allows a fine tuning of DAC audio data sampling frequency in master mode.

8.7.2.24 FCR_ADC: ADC Frequency Control Register

Register Name: FCR_ADC				Register Address: 0x17			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved	ADC_HPF	Reserved		ADC_FREQ			

Bits	Field	Description
7	Reserved	Writing has no effect, read as zero.
6	ADC_HPF	ADC High Pass Filter enable. Read/Write 0: inactive 1: enable the ADC High Pass Filter
5:4	Reserved	Writing has no effect, read as zero.
3:0	ADC_FREQ	Selection of the ADC sampling rate (Fs). Read/Write The sampling frequency value is given in the FREQ table.

NOTE: Please refer to section Sample frequency: FREQ.

8.7.2.25 CR_TIMER_MSB: counter MSB

Register Name: CR_TIMER_MSB				Register Address: 0x18			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
COUNT							

Bits	Field	Description
7:0	COUNT	MSB of counter programmable number. The counter programmable number register is split into both registers CR_TIMER_MSB and CR_TIMER_LSB.

8.7.2.26 CR_TIMER_LSB: counter LSB

Register Name: CR_TIMER_LSB				Register Address: 0x19			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
COUNT							

Bits	Field	Description
7:0	COUNT	LSB of counter programmable number The positive number decreases of '1' every slow clock period (256 MCLK periods) and generates an IRQ (TIMER_END) when going from h0001 to h0000.

8.7.2.27 ICR: Interrupt Control Register

Register Name: ICR				Register Address: 0x1A			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
INT_FORM		Reserved					

Bits	Field	Description
7:6	INT_FORM	Waveform and polarity of the IRQ signal Read/Write 00: The generated IRQ is a high level 01: The generated IRQ is a low level 10: The generated IRQ is a high level pulse with an 8 MC_CLK cycles duration when using 8-bit parallel control interface or 8 MCLK cycles duration when using SMB control interface 11: The generated IRQ is a low level pulse with an 8 MC_CLK cycles duration when using 8-bit parallel control interface or 8 MCLK cycles duration when using SMB control interface
5:0	Reserved	Writing has no effect, read as zero.

NOTE: INT_FORM should be configured to 00

8.7.2.28 IMR: Interrupt Mask Register

Register Name: IMR				Register Address: 0x1B			
bit7-RW-1	bit6-RW-1	bit5-RW-1	bit4-RW-1	bit3-RW-1	bit2-RW-1	bit1-RW-1	bit0-RW-1
ADAS_LOCK_MASK	Reserved				ADC_MUTE_MASK	Reserved	DAC_MUTE_MASK

Bits	Field	Description
7	ADAS_LOCK_MASK	Mask for the LOCK_EVENT flag Read/Write 0: interrupt enabled 1: interrupt masked (no IRQ generation)
6:3	Reserved	Writing has no effect, read as one..
2	ADC_MUTE_MASK	Mask for the ADC_MUTE_EVENT flag Read/Write 0: interrupt enabled 1: interrupt masked (no IRQ generation)
1	Reserved	Writing has no effect, read as one.
0	DAC_MUTE_MASK	Mask for the DAC_MUTE_EVENT flag Read/Write 0: interrupt enabled 1: interrupt masked (no IRQ generation)

NOTES:

1 When an interrupt is masked, the event do not generates any change on the IRQ signal, but the corresponding flag value is set to '1' in the IFR register.

2 When the IRQ signal is active on level, the IRQ signal is set to the inactive level while the bits IFR[7:0] & (!IMR[7:0]) equals '0'.

3 When the IRQ signal is a pulse, the IRQ signal is set to the inactive state until a new non-masked event occurs in IFR[7:0] or until a masked event is unmasked.

4 If the 8-bit parallel control interface is selected, MC_CLK must not be stopped in order to propagate IRQ signal.

8.7.2.29 IFR: Interrupt Flag Register

Register Name: IFR				Register Address: 0x1C			
bit7-RWC-?	bit6-RWC-?	bit5-RWC-?	bit4-R-?	bit3-R-?	bit2-RWC-?	bit1-RWC-?	bit0-RWC-?
ADAS_LOCK_EVENT	Reserved				ADC_MUTE_EVENT	Reserved	DAC_MUTE_EVENT

Bits	Field	Description
7	ADAS_LOCK_ EVENT	Lock status Read 0 = no event 1 = event detected Write 0 = no effect 1 = Reset of the flag
6:3	Reserved	Writing has no effect, read as zero.
2	ADC_MUTE_ EVENT	Read 0 = no event 1 = indicates the beginning or the end of a soft mute on ADC data Write 0 = no effect 1 = Reset of the flag
1	Reserved	Writing has no effect.
0	DAC_MUTE_ EVENT	Read 0 = no event 1 = indicates the beginning or the end of a soft mute on DAC data Write 0 = no effect 1 = Reset of the flag

NOTE: The flags RUP, RDO, GUP and GDO can be changed after 4 cycles of MCLK from codec being reset.

8.7.2.30 IMR2: Interrupt Mask Register 2

Register Name: IMR2				Register Address: 0x1D			
bit7-RW-1	bit6-RW-1	bit5-RW-1	bit4-RW-1	bit3-RW-1	bit2-RW-1	bit1-RW-1	bit0-RW-1
Reserved			TIMER_END_MASK	Reserved			

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as one.
4	TIMER_END_ MASK	Mask for the TIMER_END flag Read/Write 0: interrupt enabled 1: interrupt masked (no IRQ generation)
3:0	Reserved	Writing has no effect, read as one..

8.7.2.31 IFR2: Interrupt Mask Register

Register Name: IFR				Register Address: 0x1E			
bit7-RW-?	bit6-RW-?	bit5-RW-?	bit4-RWC-?	bit3-RW-?	bit2-RW-?	bit1-RW-?	bit0-RW-?
Reserved			TIMER_END	Reserved			

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4	TIMER_END	Timer status Read 0 = no event 1 = event detected Write 0 = no effect 1 = reset of the flag
3:0	Reserved	Writing has no effect, read as zero.

8.7.2.32 GCR_DACL: Left channel DAC Gain Control Register

Register Name: GCR_DACL				Register Address: 0x1F			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LRGOD	Reserved	GODL					

Bits	Field	Description
7	LRGOD	DAC digital gain coupling. Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6:	Reserved	Writing has no effect, read as zero.
5:0	GODL	Left channel DAC digital gain programming value.

NOTE: Please refer to section "Programmable digital attenuation: GOD" for more details.

8.7.2.33 GCR_DACR: right channel DAC Gain Control Register

Register Name: GCR_DACR				Register Address: 0x20			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved		GODR					

Bits	Field	Description
7:6	Reserved	Writing has no effect, read as zero.
5:0	GODR	Right channel DAC digital gain programming value.

NOTE: Please refer to section "Programmable digital attenuation: GOD" for more details.

8.7.2.34 GCR_DAC2L: left channel DAC2 Titanium Gain Control Register

Register Name: GCR_DAC2L				Register Address: 0x21			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-1	bit0-RW-0
Reserved				GOD2L			

Bits	Field	Description
7:6	Reserved	Writing has no effect, read as zero.
5:0	GOD2L	Left channel Line in gain programming value.

NOTE: Please refer to section "Programmable Bypass path attenuation: GI" for more details.

8.7.2.35 GCR_DAC2R: right channel DAC Mercury Gain Control Register

Register Name: GCR_LIBYR				Register Address: 0x22			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-1	bit0-RW-0
Reserved				GODR2			

Bits	Field	Description
7:6	Reserved	Writing has no effect, read as zero.
5:0	GODR2	Right channel Line in gain programming value.

NOTE: Please refer to section "Programmable Bypass path attenuation: GI" for more details.

8.7.2.36 GCR_MIC1: Microphone 1 Gain Control Register

Register Name: GCR_MIC1				Register Address: 0x23			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved					GIM1		

Bits	Field	Description
7:3	Reserved	Writing has no effect, read as zero.
2:0	GIM1	Microphone 1 boost stage gain programming value.

NOTE: Please refer to section "Programmable boost gain: GIM".

8.7.2.37 GCR_MIC2: Microphone 2 Gain Control Register

Register Name: GCR_MIC2				Register Address: 0x2F			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect, read as zero.

NOTE: Please refer to section "Programmable boost gain: GIM".

8.7.2.38 GCR_ADCL: Left channel ADC Gain Control Register

Register Name: GCR_ADCL				Register Address: 0x25			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LRGID	Reserved	GIDL					

Bits	Field	Description
7	LRGID	ADC digital gain coupling. Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6	Reserved	Writing has no effect, read as zero.
5:0	GIDL	Left channel ADC digital gain programming value.

NOTE: Please refer to the section "Programmable input attenuation amplifier: GID".

8.7.2.39 GCR_ADCR: Right channel ADC Gain Control Register

Register Name: GCR_ADCR				Register Address: 0x26			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect, read as zero.

NOTE: Please refer to the section "Programmable input attenuation amplifier: GID".

8.7.2.40 GCR_MIXDACL: DAC Left Digital Mixer Control Register

Register Name: GCR_MIXDACL				Register Address: 0x27			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LRGOMIX	Reserved	GOMIXL					

Bits	Field	Description
7	LRGOMIX	Mixer gain coupling for DAC path Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6	Reserved	Writing has no effect, read as zero.
5:0	GOMIXL	Mixer gain for DAC path. Read/Write 000000 : 0dB 000001 : -1dB …by step of 1dB 111111 : -31dB

8.7.2.41 GCR_MIXDACR: DAC Right Digital Mixer Control Register

Register Name: GCR_MIXDACR				Register Address: 0x28			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved		GOMIXR					

Bits	Field	Description
7:6	Reserved	Writing has no effect, read as zero.
5:0	GOMIXR	Mixer gain for DAC path. Read/Write 000000 : 0dB 000001 : -1dB …by step of -1dB 111111 : -31dB

8.7.2.42 GCR_MIXADCL: ADC Left Digital Mixer Control Register

Register Name: GCR_MIXADCL				Register Address: 0x29			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LRGIMIX	Reserved	GIMIXL					

Bits	Field	Description
7	LRGIMIX	Mixer gain coupling for input path Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6	Reserved	Writing has no effect, read as zero.
5:0	GIMIXL	Mixer gain for input path. Read/Write 000000 : 0dB 000001 : -1dB … by step of -1dB 111111 : -31dB

8.7.2.43 GCR_MIXADCR: ADC Right Digital Mixer Control Register

Register Name: GCR_MIXADCR				Register Address: 0x2A			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved		GIMIXR					

Bits	Field	Description
7:6	Reserved	Writing has no effect, read as zero.
5:0	GIMIXR	Mixer gain for input path. Read/Write 000000 : 0dB 000001 : -1dB … by step of -1dB 111111 : -31dB

8.7.2.44 CR_DAC_AGC: DAC Soft clipping DRC Control Register

Register Name: CR_DAC_AGC				Register Address: 0x2B			
bit7-RW-0	bit6-RW-0	bit5-RW-1	bit4-RW-1	bit3-RW-0	bit2-RW-1	bit1-RW-0	bit0-RW-0
DAC_AGC_EN	DAC_AGC_ LOAD	DAC_AGC_ADD					

Bits	Field	Description
7	DAC_AGC_EN	Enable of the AGC system. Read/Write 0 : inactive 1 : enable the automatic level control
6	DAC_AGC_LOAD	Read/Write

		Refer to DR_ADC_AGC for ADC_AGC_LOAD description
5:0	DAC_AGC_ADD	Address of AGC control registers Read/Write: DAC_AGC_ADD[5:0] / Corresponding control register 000000 / DAC_AGC_0 000001 / DAC_AGC _1 000010 / DAC_AGC _2 000011 / DAC_AGC _3 000100 / DAC_AGC _4

8.7.2.45 DR_DAC_AGC: DAC Soft clipping DRC Data Register

Register Name: DR_DAC_AGC				Register Address: 0x2C			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
DAC_AGC_DATA							

Bits	Field	Description
6	DAC_AGC_DATA	DAC Soft clipping DRC indirect access data register.

8.7.2.45.1 DAC_AGC_0: DAC Soft clipping DRC control register 0

Register Name: DAC_AGC_0				Indirect register address: 0x00			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LR_DRC	Reserved			LTHRES			

Bits	Field	Description
7	LR_DRC	DRC parameter coupling 0: Left and right channels DRC parameters are independent 1: Left and right channels DRC parameters are the same: LTHRES and LCOMP are used for both left and right channels
6:5	Reserved	Writing has no effect, read as zero.
4:0	LTHRES	DRC compression threshold for left channel DAC 00000: Full scale 00001: Full scale -1dB 00010: Full scale -2dB by step of - dB 11110: Full scale -30dB 11111: Full scale -31dB

8.7.2.45.2 DAC_AGC_1: DAC Soft clipping DRC control register 1

Register Name: DAC_AGC_1				Indirect register address: 0x01			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved					LCOMP		

Bits	Field	Description
7:3	Reserved	Writing has no effect, read as zero.
2:0	LTHRES	DRC compression rate for left channel DAC 000: 1 001: 2 step x2 100: 16 101: 32 Higher than 101: Reserved for Dolphin's usage

8.7.2.45.3 DAC_AGC_2: DAC Soft clipping DRC control register 2

Register Name: DAC_AGC_2				Indirect register address: 0x02			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved			RTHRES				

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	LTHRES	DRC compression threshold for right channel DAC 00000: Full scale 00001: Full scale -1dB 00010: Full scale -2dB by step of - dB 11110: Full scale -30dB 11111: Full scale -31dB

8.7.2.45.4 DAC_AGC_3: DAC Soft clipping DRC control register 3

Register Name: DAC_AGC_3				Indirect register address: 0x03			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved					RCOMP		

Bits	Field	Description
7:3	Reserved	Writing has no effect, read as zero.
2:0	RCOMP	DRC compression rate for right channel DAC 000: 1 001: 2 step x2 100: 16 101: 32 Higher than 101: Reserved for Dolphin's usage

8.7.2.46 CR_DAC2_AGC: DAC2 Soft clipping DRC Control Register

Register Name: CR_DAC2_AGC: DAC2				Register Address:ox2D			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect, read as zero.

8.7.2.47 DR_DAC2_AGC: DAC2 Soft clipping DRC Data Register

Register Name: DR_DAC2_AGC				Register Address: 0x2E			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-1	bit0-RW-1
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect, read as zero.

8.7.2.47.1 DAC2_AGC_0: DAC2 Soft clipping DRC control register

Register Name: DAC2_AGC_0				Indirect register address: 0x00			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LR_DRC	Reserved			LTHRES			

Bits	Field	Description
7	LR_DRC	DRC parameter coupling 0: Left and right channels DRC parameters are independent 1: Left and right channels DRC parameters are the same: LTHRES and LCOMP are used for both left and right channels
6:5	Reserved	Writing has no effect, read as zero.

4:0	LTHRES	DRC compression threshold for left channel DAC 00000: Full scale 00001: Full scale -1dB 00010: Full scale -2dB by step of - dB 11110: Full scale -30dB 11111: Full scale -31dB

8.7.2.47.2 DAC2_AGC_1: DAC2 Soft clipping DRC control register 1

Register Name: DAC2_AGC_1				Indirect register address: 0x01			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved					LCOMP		

Bits	Field	Description
7:3	Reserved	Writing has no effect, read as zero.
2:0	LTHRES	DRC compression rate for left channel DAC 000: 1 001: 2 step x2 100: 16 101: 32 Higher than 101: Reserved for Dolphin's usage

8.7.2.47.3 DAC2_AGC_2: DAC Soft clipping DRC control register 2

Register Name: DAC2_AGC_2				Indirect register address: 0x02			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved			RTHRES				

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	LTHRES	DRC compression threshold for right channel DAC 00000: Full scale 00001: Full scale -1dB 00010: Full scale -2dB by step of - dB 11110: Full scale -30dB 11111: Full scale -31dB

8.7.2.47.4 DAC2_AGC_3: DAC Soft clipping DRC control register 3

Register Name: DAC2_AGC_3				Indirect register address: 0x03			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved					RCOMP		

Bits	Field	Description
7:3	Reserved	Writing has no effect, read as zero.
2:0	RCOMP	DRC compression rate for right channel DAC 000: 1 001: 2 step x2 100: 16 101: 32 Higher than 101: Reserved for Dolphin's usage

8.7.2.48 CR_ADC_AGC: Automatic Gain Control Register

Register Name: CR_ADC_AGC				Register Address: 0x2F			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
ADC_AGC_EN	ADC_AGC_ LOAD	ADC_AGC_ADD					

Bits	Field	Description
7	ADC_AGC_EN	Enable of the AGC system on ADC path Read/Write 0 : inactive 1 : enable the automatic level control
6	ADC_AGC_LOAD	ADC Automatic Gain Control indirect access load bit.
5:0	DAC_AGC_ADD	ADC Automatic Gain Control indirect access address.

8.7.2.49 DR_ADC_AGC: Automatic Gain Control Data Register

Register Name: DR_ADC_AGC				Register Address: 0x30			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
ADC_AGC_DATA							

Bits	Field	Description
7:0	ADC_AGC_DATA	data loaded in AGC control registers Read/Write ● Instructions for writing into a register The data must firstly be written into the DR_ADC_AGC register. The register address must then be written in CR_ADC_AGC

		(ADC_AGC_ADD[5:0]). The data is written into the register when ADC_AGC_LOAD is set to '1'. ● Instructions for reading the values of a register The address of the register to be read is written in CR_ADC_AGC (ADC_AGC_ ADD[5:0]). The content of the corresponding register appears in DR_ADC_AGC when ADC_AGC_LOAD is set to '0'.

8.7.2.49.1 ADC_AGC_0: AGC control register 0

Register Name: ADC_AGC_0				Indirect register Address:0x00			
bit7-RW-0	bit6-RW-0	bit5-RW-1	bit4-RW-1	bit3-RW-0	bit2-RW-1	bit1-RW-0	bit0-RW-0
Reserved		TARGET				Reserved	

Bits	Field	Description
7:6	Reserved	Writing has no effect, read as zero.
5:2	TARGET	Target output level of the ADC Read/Write: 0000 = -6 dB 0001 = -7.5 dB … by step of 1.5 dB 1111 = - 28.5 dB
1:0	Reserved	Writing has no effect, read as zero.

8.7.2.49.2 ADC_AGC_1: Automatic Gain Control Register 2

Register Name: ADC_AGC_1				Indirect register Address:0x01			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-1	bit0-RW-1
NG_EN	NG_THR			HOLD			

Bits	Field	Description
7	NG_EN	Selection of the Noise Gate system. Read/Write 0: inactive 1: enables the noise gate system
6:4	NG_THR	Noise Gate Threshold value. Input level (dB) < Noise Gate Level (dB). Read/Write 000: -72 dB 001: -66 dB …by step of 6dB 111: -30 dB
3:0	HOLD	Hold time before starting AGC adjustment to the TARGET value. Read/Write

		0000: 0ms 0001: 2 ms 0010: 4 ms … Time Step x2 1111: 32.768s

NOTE: Please refer to section "AGC system guide" for more details.

8.7.2.49.3 ADC_AGC_2: AGC control register 2

Register Name: ADC_AGC_2				Indirect register Address:0x02			
bit7-RW-0	bit6-RW-1	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-0	bit0-RW-0
ATK				DCY			

Bits	Field	Description
7:4	ATK	Attack Time - Gain Ramp Down. Read/Write 0000: 32 ms 0001: 64 ms …by step of 32 ms 1111: 512 ms
3:0	DCY	Decay Time - Gain Ramp up. Read/Write 0000: 32 ms 0001: 64 ms …by step of 32 ms 1111: 512 ms

NOTES:

1 DCY and ATK registers values are delays between each step of gain.

2 Please refer to section "AGC system guide" for more details.

8.7.2.49.4 ADC_AGC_3: AGC control register 3

Register Name: ADC_AGC_3				Indirect register Address: 0x03			
Bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-1	bit3-RW-1	bit2-RW-1	bit1-RW-1	bit0-RW-1
Reserved				AGC_MAX			

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	AGC_MAX	Maximum Gain Value to apply to the ADC path.

NOTES:

1 Please refer below table for AGC_MAX setup.

2 Please refer to section "AGC system guide" for more details.

AGC_MAX	Gain Value	AGC_MAX	Gain Value	AGC_MAX	Gain Value	AGC_MAX	Gain Value
00000	0	01000	12	10000	23	11000	32
00001	1.5	01001	13.5	10001	23	11001	33.5
00010	3	01010	15	10010	23	11010	35
00011	4.5	01011	16.5	10011	24.5	11011	36.5
00100	6	01100	18	10100	26	11100	38
00101	7.5	01101	19.5	10101	27.5	11101	39.5
00110	9	01110	21	10110	29	11110	41
00111	10.5	01111	22.5	10111	30.5	11111	42.5

8.7.2.49.5 ADC_AGC_4: AGC control register 4

Register Name: ADC_AGC_4				Indirect register Address: 0x04			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved				AGC_MIN			

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	AGC_MIN	Maximum Gain Value to apply to the ADC path.

NOTES:

1 Please refer to below table for AGC_MIN setup.

AGC_MIN	Gain Value	AGC_ MIN	Gain Value	AGC_ MIN	Gain Value	AGC_ MIN	Gain Value
00000	0	01000	12	10000	23	11000	32
00001	1.5	01001	13.5	10001	23	11001	33.5
00010	3	01010	15	10010	23	11010	35
00011	4.5	01011	16.5	10011	24.5	11011	36.5
00100	6	01100	18	10100	26	11100	38
00101	7.5	01101	19.5	10101	27.5	11101	39.5
00110	9	01110	21	10110	29	11110	41
00111	10.5	01111	22.5	10111	30.5	11111	42.5

2 Please refer to section "AGC system guide" for more details.

8.7.2.50 SR_ADC_AGCDGL: ADC AGC left digital gain

Register Name: SR_ADC_AGCDGL				Register Address:0x31			
bit7-RW-?	bit6-RW-?	bit5-RW-?	bit4-RW-?	bit3-RW-?	bit2-RW-?	bit1-RW-?	bit0-RW-?
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect.

8.7.2.51 SR_ ADC_AGCDGR: ADC AGC right digital gain

Register Name: SR_ ADC_AGCDGR				Register Address:0x32			
bit7-RW-?	bit6-RW-?	bit5-RW-?	bit4-RW-?	bit3-RW-?	bit2-RW-?	bit1-RW-?	bit0-RW-?
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect.

8.7.2.52 SR_ADC_AGCDGL: ADC AGC left analog gain

Register Name: SR_ ADC_AGCDGL				Register Address:0x33			
bit7-RW-?	bit6-RW-?	bit5-RW-?	bit4-RW-?	bit3-RW-?	bit2-RW-?	bit1-RW-?	bit0-RW-?
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect.

8.7.2.53 SR_ADC_AGCDGR: ADC AGC right analog gain

Register Name: SR_ ADC_AGCDGR				Register Address:0x34			
bit7-RW-?	bit6-RW-?	bit5-RW-?	bit4-RW-?	bit3-RW-?	bit2-RW-?	bit1-RW-?	bit0-RW-?
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect.

8.7.2.54 CR_TR: Test Mode Control Register

Register Name: CR_TR				Register Address:0x35			
bit7-RW-?	bit6-RW-?	bit5-RW-?	bit4-RW-?	bit3-RW-?	bit2-RW-?	bit1-RW-?	bit0-RW-?
FAST_ON	Reserved						

Bits	Field	Description
7	FAST_ON	Disables internal pop reduction mechanisms for quicker start-up Must be put to '0' to ensure correct CODEC performances
6:0	Reserved	Writing has no effect.

8.7.2.55 DR_TR: Test Mode Data Register

Register Name: DR_TR				Register Address:0x36			
bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect, read as zero.

8.7.2.56 SR_TR1: Test Mode Status Register

Register Name: SR_TR1				Register Address:0x37			
bit7-RW-?	bit6-RW-?	bit5-RW-?	bit4-RW-?	bit3-RW-?	bit2-RW-?	bit1-RW-?	bit0-RW-?
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect.

8.7.2.57 SR_TR2: Test Mode Status Register 2

Register Name: SR_TR2				Register Address:0x38			
bit7-RW-?	bit6-RW-?	bit5-RW-?	bit4-RW-?	bit3-RW-?	bit2-RW-?	bit1-RW-?	bit0-RW-?
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect.

8.7.2.58 SR_TR_SRCDAC: DAC test Mode Status Register

Register Name: SR_TR_SRCDAC				Register Address:0x39			
bit7-RW-?	bit6-RW-?	bit5-RW-?	bit4-RW-?	bit3-RW-?	bit2-RW-?	bit1-RW-?	bit0-RW-?
Reserved							

Bits	Field	Description
7:0	Reserved	Writing has no effect.

8.8 Programmable gains

This section helps you to configure the programmable gain amplifier in the CODEC.
Internal signal VREFP is connected to AVDCDC Pin and internal signal VREFN is connected to CODEC_AVSS Pin.

In this section, VREF equals to (VREFP – VREFN).

8.8.1 Programmable boost gain: GIM

The following table gives the relation between the gain and the input level for the microphone input amplifier **when GID = 0.**

GIM[2]	GIM[1]	GIM[0]	Gain value (dB)	Maximum input amplitude (Vp-p differential) (FS)
0	0	0	0	0.85*VREF
0	0	1	4	0.536*VREF
0	1	0	8	0.338*VREF
0	1	1	12	0.213*VREF
1	0	0	16	0.134*VREF
1	0	1	20	0.085*VREF
1	1	0	20	0.085*VREF
1	1	1	20	0.085*VREF

NOTES:

1 **Analog input amplitude value is not more than 'Maximum input amplitude' for Vp-p differential.**

2 This maximum analog input amplitude is referenced as Full Scale (FS). After conversion, the corresponding digital code of the output value varies from 0x7FFF down to 0x8000 for a 16-bit word. When the analog input amplitude value is greater than FS, the dynamic characteristics are not guaranteed.

3 When a change occurs on GIDi inputs, data are valid on the digital output after about 64 sample periods. If the HPF is activated, data are valid after about 64 sample periods but the offset cancellation is not still completed at this time due to its internal time constant.

8.8.2 Programmable input gain amplifier: GID

The digital gain of ADC path may be programmed through the registers bits GIDL and GIDR.
The value of the gain is programmable from 0 to 23dB with a pitch of 1dB.

The gain and input levels are obtained according to the following table:

GID						Decimal decoded value	Gain value (dB)	Maximum input amplitude (Vp-p. Differential) (FS)
0	0	0	0	0	0	0	0	0.85*VREF
0	0	0	0	0	1	1	1	0.757*VREF
0	0	0	0	1	0	2	2	0.6021*VREF
…								
x	y	z	t	u	v	i	i	0.85/{10^(i/20)}*VREF
…								
0	1	0	1	1	1	23	23	0.06*VREF
0	1	1	0	0	0	24	24	0.06*VREF
…								
1	0	1	0	1	0	43	43	0.006*VREF

NOTE: The last column of the table shows the maximum analog input to **be applied to** the MICi inputs. The value is given in Vp-p differential. These values refer to the external voltage reference VREF **equal** to (VREFP – VREFN). The voltage levels depend on the VREF voltage.

8.8.3 Programmable digital attenuation: GOD

The attenuation of DAC output amplifier may be programmed independently for the both channels through the registers bits GODL and GODR.

The value of the gain GODL/R is programmable from +0 to –31dB with 1 dB pitch. The gain and output levels are obtained according to the following table:

GOD					Decimal decoded value	Gain Value (dB)
0	0	0	0	0	0	0
0	0	0	0	1	1	-1
					…	
0	0	1	1	0	6	-6

					…	
1	1	1	1	0	30	-30
1	1	1	1	1	31	-31

8.8.4 Programmable attenuation: GO

The attenuation of Headphone output amplifier may be programmed independently for the both channels through the registers bits GOL and GOR.

The value of the gain GOL/R is programmable from +6 to –25dB with 1 dB pitch. The gain and output levels are obtained according to the following table:

GO					Decimal decoded value	Gain Value (dB)	Maximal PGAT input amplitude (Vp-p)	Maximal PGAT output amplitude (Vp-p)
0	0	0	0	0	0	+6	0.425*VREF	0.85*VREF
0	0	0	0	1	1	+5	0.478*VREF	0.85*VREF
					…		…	…
0	0	1	0	1	5	+1	0.757*VREF	0.85*VREF
0	0	1	1	0	6	0	0.85*VREF	0.85*VREF
0	0	1	1	1	7	-1	0.85*VREF	0.757*VREF
					…		…	…
1	1	1	1	0	30	-24	0.85*VREF	0.054*VREF
1	1	1	1	1	31	-25	0.85*VREF	0.048*VREF

NOTES:

1 When headphone driver is loaded by a 16 Ohm load, setting GOL/R = 0 is possible. However, set GOL/R to 9 at the maximum to preserve dynamic performances. The output stage is sized to support a 70mA current at most.

2 The last column of the table shows the analog output voltage in the condition of a digital input at FS (Full Scale). The value is given in Vp-p single-ended.

3 These values refer to the external voltage reference VREF equal to (VREFP – VREFN). The voltage levels depend on the VREF voltage.

8.8.5 Programmable digital mixer gain: GIMIX and GOMIX

The following table shows the relation between the gain and GIMIX/GOMIX.

GIMIX or GOMIX	Gain value (dB)
00000	0

00001	-1
00010	-2
00011	-3
…	…
11101	-29
11110	-30
11111	-31

8.8.6 Gain refresh strategy

GI* and GO* gains are controlled through the control interface. To avoid sound artifacts, the gain increases or decreases each time the gain stage output crosses the zero value. **Tcrossout** time-out counter forces the gain to be updated if a zero crossing event doesn't occur. After each gain step, zero crossing events are ignored during at least **Tcrossmin**.

In case that the gain coupling between both left and right channels is active (LRGi different from RLGi), gain stepping of each channel is independent from the other depending on zero crossing event occurrence.

The duration of Tcrossout and Tcrossmin are given below:

MCLK (MHz)	**Tcrossout (ms)**	**Tcrossmin (ms)**
12	21.8	0.171
13	20.2	0.158

8.9 Configuration of the headphone output stage

In cap-coupled connection, codec uses the LOAD register bit to control the ramping duration. Inappropriate setting will lead to a too long or too fast ramping and will create audio artifacts.

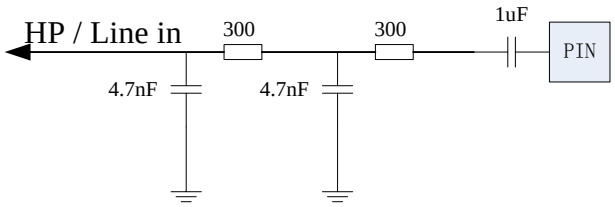
To prevent pop-up noise generation due to floating nodes when no load is plugged in the jack connector, it is required to add some resistor devices that act as pull down function (named Rhpl and Rhpr in section "Headphone connnection" and section "Required external components").

Its value has to be determined as following:

Working Mode	**Load resistor and bypass capacitor values**	**LOAD value**	**Rhpdo value**
Driving Headphone	16 Ohm / 220uF	0	470 Ohm typ.
Driving Lineout	10k Ohm / 1uF	1	4.7k Ohm max.

8.10 Out-of-band noise filtering

Reduction circuit (Low Pass Filter) at PCB after the DAC output is used to remove the out-of-band noise generated by the delta sigma modulation (Noise Shaper). The circuit reduces the amount of energy contained in the wide band part (> 24 kHz) of the output signal. The out-of-band noise, when not removed, can be damageable in some high quality applications.



8.10.1 Reset of short circuit detection

The following sequence has to be to apply:

1 Mask the interrupt by writing ‘1’ in the Interrupt Control Register.

2 Handle the cause of short-circuit according to the events presented in following paragraphs (Capacitor-coupled headphone connection).

3 Update the short-circuit flag by writing ‘1’ in the Interrupt Flag Register.

4 Check the reset of flag by reading the Interrupt Flag Register. The bit must be equal to ‘0’. If it remains at '1', that means that short-circuit is not resolved.

5 Enable the interrupt by writing ‘0’ in the Interrupt Control Register.

8.11 Sampling frequency: FREQ

The sampling frequency value is given in the FREQ table below.

FREQ[3:0]	Sampling Rate (Fs)
0000	8 kHz
0001	11.025 kHz
0010	12 kHz
0011	16 kHz
0100	22.05 kHz
0101	24 kHz
0110	32 kHz
0111	44.1 kHz

1000	48 kHz
1001	88.2 kHz
1010	96 kHz
1011	176.4 kHz
1100	192 kHz
1101	Reserved
1110	Reserved
1111	Reserved

8.12 Programmable data word length

The Data Word Length block (DWL) allows selecting the length of the input data and of the output data between 24-/20-/18-/16-bit **to AICR_DAC. DAC_ADWL and AICR_ADC.ADC_ADWL** (respectively for the DAC and ADC paths) in accordance with the following table:

ADWL	Word length
0 0	16-bit word length data
0 1	18-bit word length data
1 0	20-bit word length data
1 1	24-bit word length data

The size of the buses is always 24 bits, but the input/output data only use the number of MSB programmed with ADWL. The LSB are considered as ‘0’ in input and set to ‘0’ in output.

The capability to use a data word length of 16 bits is kept for compatibility with standard applications.

8.13 Ramping system note

An internal mechanism is used to reduce output glitches when the headphone stage enters or leaves the power-down mode.

When the SB_HP is set to ‘1’, the headphone output voltages (AOHPL, AOHPR) are slowly decreased in the same time from /2 down to 0.

When the SB_HP is set to ‘0’, the headphone output voltages (AOHPL, AOHPR) are slowly increased in the same time from 0 to VREFP/2.

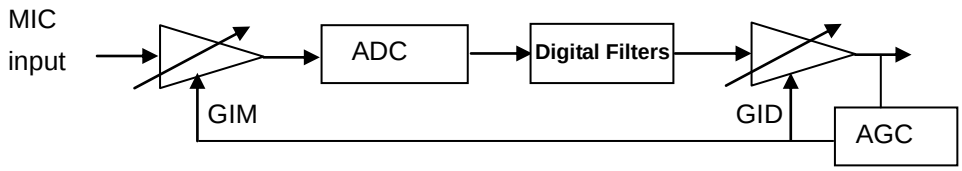
After power supplies ramp up, the CODEC starts its internal initialization sequence and SR. PON_ACK register is changed when the initialization sequence is complete.
An interrupt request is sent when the ramp completes.

Do not change the level of SB_HP as long as the **initialization** sequence is not complete. If the change occurs, the success of the **initialization** sequence is not **guaranteed.**
In order to prevent audible glitch, it is required to power-down the output stage (SB_HP=1) before changing the output load with CR1.LOAD.

Please refer to "Anti-pop operation sequences" for details.

8.14 AGC system guide

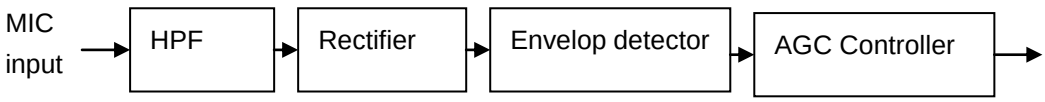
For the microphone input to ADC path, an Automatic Gain Control (AGC) system allows to optimize the signal swing at the input of the ADC.



The AGC circuit compares the output of the ADC to a level and increases or decreases the gain of the microphone preamplifier and the digital gain to compensate. The full dynamic range of the ADC can be used automatically if the audio from the microphone is to be output digitally through the ADC.

The AGC_EN register bit enables the AGC system, in this case IN_SEL must be equal to "00" or "01".

AGC Block Diagram:



The AGC system is used at the MIC input.
The HPF filter characteristics: Cut Frequency =300 Hz.
In the in AGC mode, the system of gain control will directly assign the values of the gains GIDL, GIDR and GIM1 (or GIM2).

8.14.1 AGC operating mode

TARGET sets the desired ADC output range level. The AGC system adapts the gain stages (GID and GIM) in order to best reach this target. AGC_MAX and AGC_MIN fix the limits of the gain variation.
Please refer to "CODEC Operating modes" for the AGC System diagram in the "CODEC Power Diagram".

In order that the AGC system should not alter the dynamic content of the signal (voice “tonic” for instance) by continuously adapting the gain to fit the target level, the time between two consecutives gain adjustments is modifiable by the HOLD register value.

After this delay:

– If the output level is lower than TARGET, the gain is increased step by step in accordance to the DCY register value.

– If the output level is higher than TARGET, the gain is decreased step by step in accordance to the ATK register value.

The following figure illustrates the behavior of AGC system:

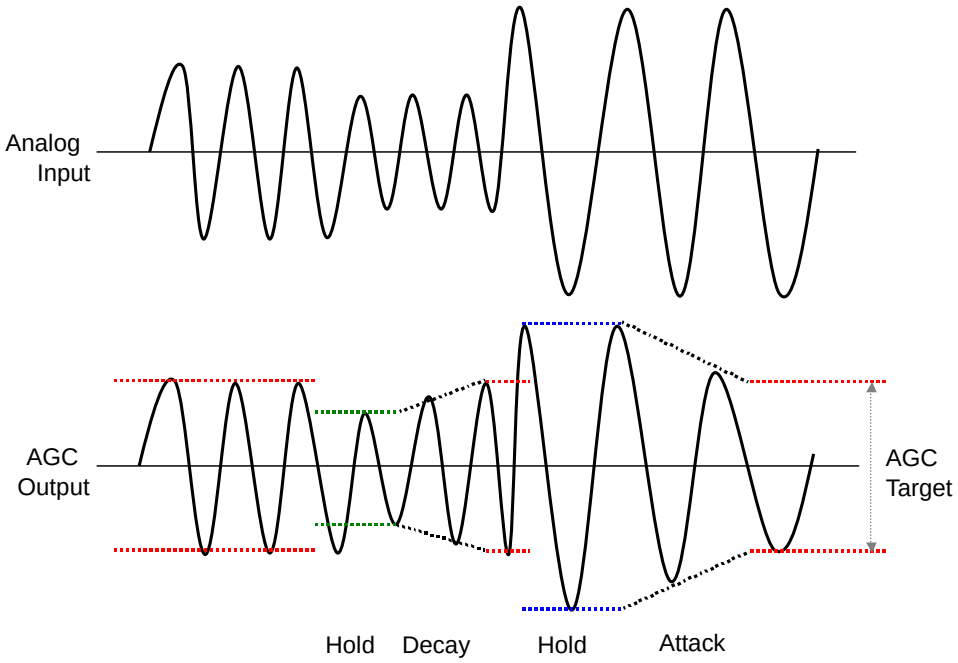


Figure 8-3 AGC adjusting waves

A noise-gating feature, enabled by the NG_EN register bit, prevents gain increases when no signal or small signal is present at the input. The noise gate threshold is set by the NG_THR register value. The following graph shows a more detailed application.

The following graph summarizes the operations and shows more details.

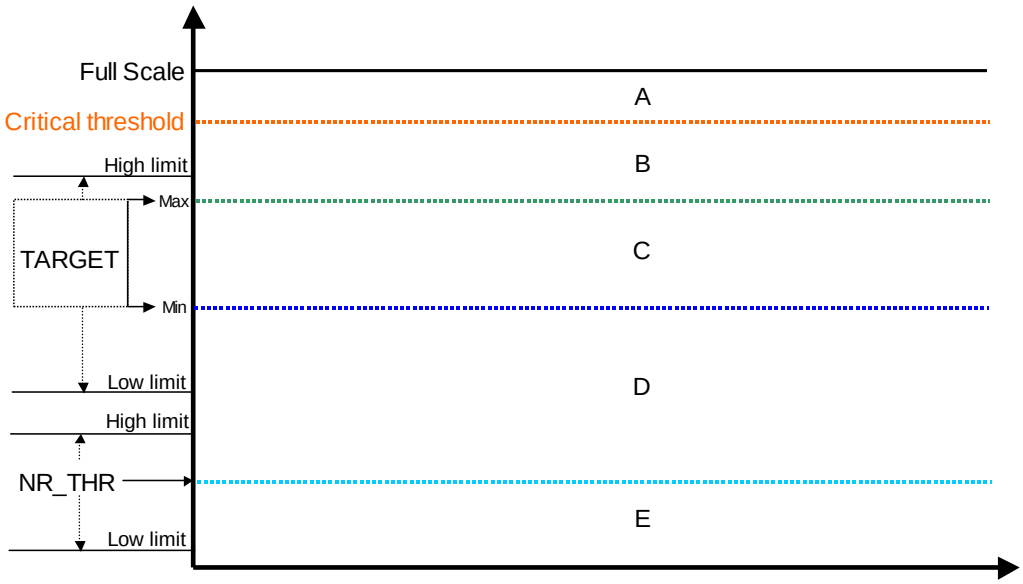


Figure 8-4 AGC adjust areas

The areas from A to E are **different working areas** of AGC system, which is listing below:

A: If the signal level is in this critical area: the AGC system decreases quickly the gain at the input of the ADC until the signal goes under the critical threshold.

B: If the signal level remains in this area after the HOLD delay: the AGC system decreases the gain at the input of the ADC until the signal reaches the target area with a slope defined by AGC3.ATK register value.

C: If the signal level is in this area: the AGC system does not perform gain adjustment.

D: If the signal level remains in this area after the HOLD delay: the AGC system increases gain at the input of the ADC until the signal reach the target area with a slope defined by AGC3.DCY register value.

E: If the signal level is in this range: the AGC system considers the signal as noise and does not perform gain adjustment.

8.15 DRC description

The Soft Clipping DRC is a digital signal soft limiter. Its goal is to limit digital signal when amplification gain is applied in the digital domain (GOD* > 0dB), to reduce occurence of possible hard clipping event when input signal reach fullscale. This function is enabled by setting the bit DAC_AGC_EN to ’1’ and disable by setting ’0’. When enabled, signals at the input of DAC noise shaper is soft clipped when they are larger than the compression threshold (programmed in soft clipping registers DAC_AGC_*).

The part above the threshold id divided by the compression rate (programmed in soft clipping registers DAC_AGC_*); thus, the signal above threshold at the output of the soft clipping DRC is:

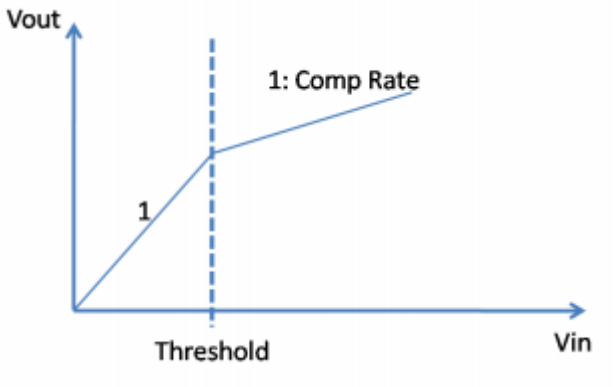


Figure 8-5 signal above threshold at the output of the soft clipping DRC

$$SCDRC_{input} = \frac{\left(DAC_{input} \times GOMIX* \times GOD*\right) - Threchold}{CompRate} + Threshold$$

8.16 Digital Mixer description

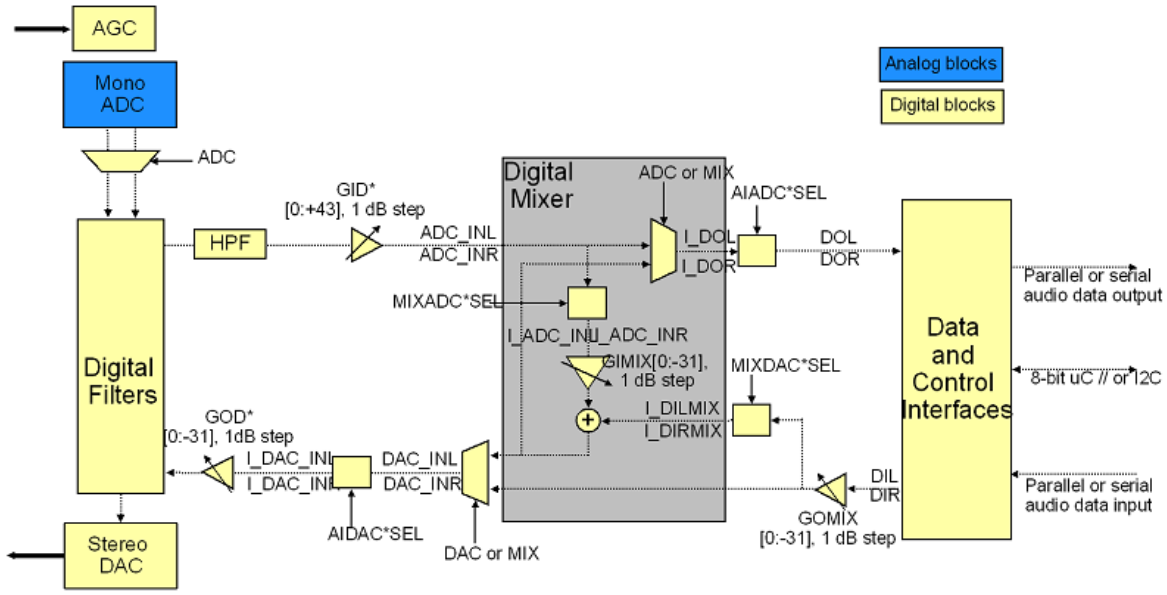


Figure 8-6 Digital Mixer structure

A digital multiplexer allows choosing between the ADC signal and the mixer output signal on the record path. Another digital multiplexer allows choosing between the DAC signal and the mixer output signal on the playback path.

AIDAC*SEL, AIADC*SEL, MIXDAC*SEL and MIXADC*SEL functionality are applied between the Digital Mixer and the Data interface. The behavior is the following:

AIDACLSEL1	AIDACLSEL0	I_ DAC_INL	AIDACRSEL1	AIDACRSEL0	I_ DAC_INR
0	0	DAC_INR	0	0	DAC_INL
0	1	DAC_INL	0	1	DAC_INR
1	0	(DAC_INL + DAC_INR)/2	1	0	(DAC_INL + DAC_INR)/2
1	1	0	1	1	0

AIADCLSEL1	AIADCLSEL0	DOL	AIADCRSEL1	AIADCRSEL0	DOR
0	0	I_DOR	0	0	I_DOL
0	1	I_DOL	0	1	I_DOR
1	0	(I_DOL+I_DOR)/2	1	0	(I_DOL+I_DOR)/2
1	1	0	1	1	0

MIXDACLSEL1	MIXDACLSEL0	I_DILMIX	MIXDACRSEL1	MIXDACRSEL0	I_DIRMIX
0	0	DIL	0	0	DIR
0	1	DIR	0	1	DIL
1	0	(DIL+DIR)/2	1	0	(DIL+DIR)/2
1	1	0	1	1	0

MIXADCLSEL1	MIXADCLSEL 0	I_ADC_INL	MIXADCRSEL1	MIXADCRSEL 0	I_ADC_INR
0	0	ADC_INR	0	0	ADC_INL
0	1	ADC_INL	0	1	ADC_INR
1	0	(ADC_INL+ADC_INR)/2	1	0	(ADC_INL+ADC_INR)/2
1	1	0	1	1	0

8.17 Digital microphone interface

CODEC accepts bitstream from digital microphone and converts it into audio data at the sample rate (Fs) selected in FCR_ADC register. CODEC provides a clock (DMIC_CLK) and receives data on

DMIC_IN at the same frequency. DMIC_CLK frequency depends on MCLK frequency selection in CCR register.

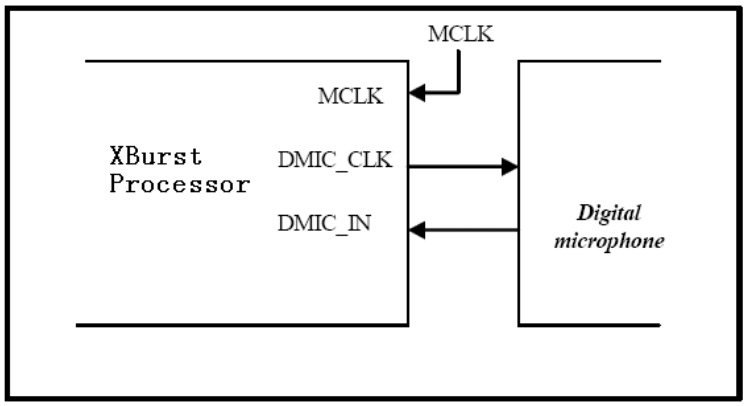


Figure 8-7 Digital microphone interface connection

After conversion, the corresponding digital code of the output value varies from 0x7FFF down to 0x8000 for a 16-bit word, coded in 2's complement.

CODEC can receive simultaneously data from two digital microphones.

8.17.1 Timing Diagram

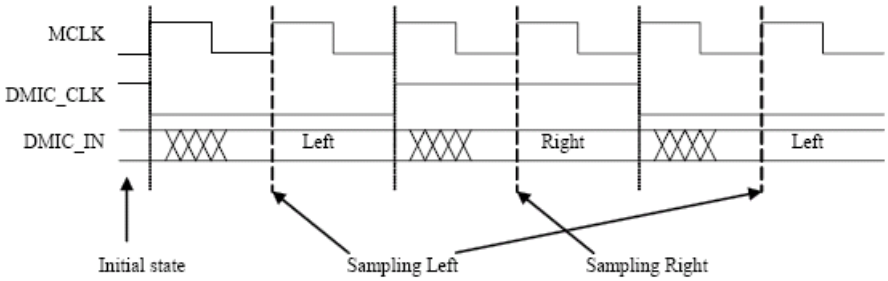
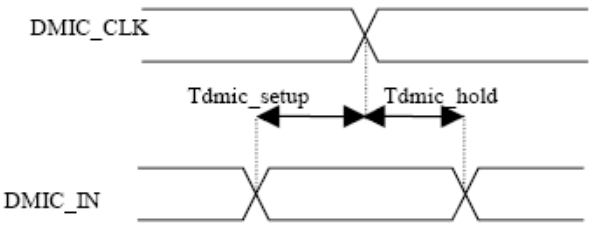


Figure 8-8 Digital microphone timing diagram at MCLK = 12 MHz (DMIC_CLK = 3 MHz)

8.17.2 Timings



Parameter	Symbol	Min	Typ	Max	Unit
DMIC_CLK frequency	F_{dmic_clk}	3	-	3.25	MHz
DMIC_CLK duty cycle	D_{dmic_clk}	0.4	0.5	0.6	-
DMIC_IN setup time	T_{dmic_setup}	T_{MCLK} + 10	-	-	ns
DMIC_IN hold time	T_{dmic_hold}	0	-	-	ns

8.17.3 Noise template (TBC)

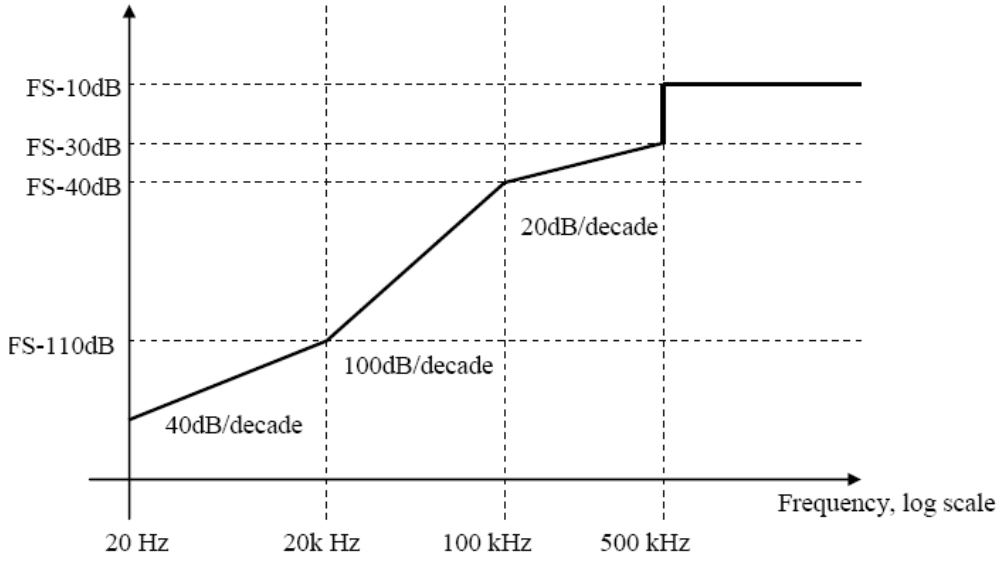


Figure 8-9 Digital microphone modulation noise reference spectrum (with FFT resolution = 20 Hz and 7 terms Blackman-Harris windowing)

If the noise at the digital microphone output is higher than the reference in the [20 Hz – 20 kHz] bandwidth, the SNR will be limited by the digital microphone in-band noise.

If the noise at the digital microphone output is higher than the reference for frequencies beyond 20 kHz, the SNR will be limited by the aliasing of the digital microphone quantization noise.

8.17.4 Power Supply Noise Tolerance Template (PSNT2) on analog power supply for I/O buffers of DAC outputs (VDDIO_CODEC)

The following graphic represents the maximum noise allowed on CODEC_AVDD to reach SNR performances of 80 or 95 or 105 dB on the DAC outputs.

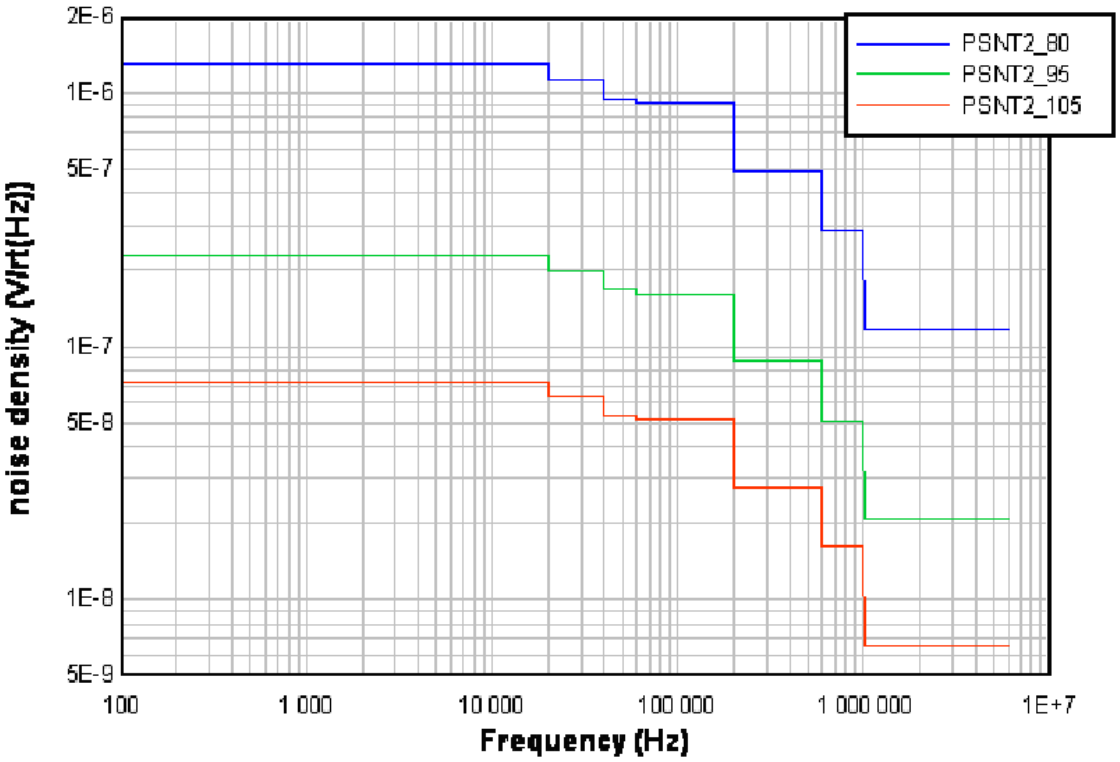


Figure 8-10 PSNT2 for VDDIO_CODEC when using PWM output

Frequency(Hz) Noise density (μV RMS)	10-20k	20k-40k	40-60k	60k-200k	200k-600k	600k-1M	1M-6.3M
PSNT2_80dB	182	158	134	339	308.9	181	267.7
PSNT2_95dB	32.4	28.2	24	60.4	54.9	32	47.6
PSNT2_105dB	10.2	8.9	7.6	19.1	17.4	10.2	15.1

Figure 8-11 PSNT2 for VDDIO_CODEC when using PWM output

8.18 CODEC Operating modes

Different operating modes are available:

– Power-up mode: During power on time, CODEC is in this mode.
– Reset mode: When NRST is low, CODEC is in this mode.
– Soft mute mode: When DAC_MUTE is 1, CODEC is in this mode.
– Complete Power-down mode: After RESET, CODEC is in this mode.
– SLEEP modes: When SB_SLEEP is 1, CODEC is in this mode.
– Normal mode: When CODEC is not in above mode, it is in this mode. This mode has three modes: RECORD mode, REPLAY mode, RECORD_REPLAY mode.

The power diagram is shown below.

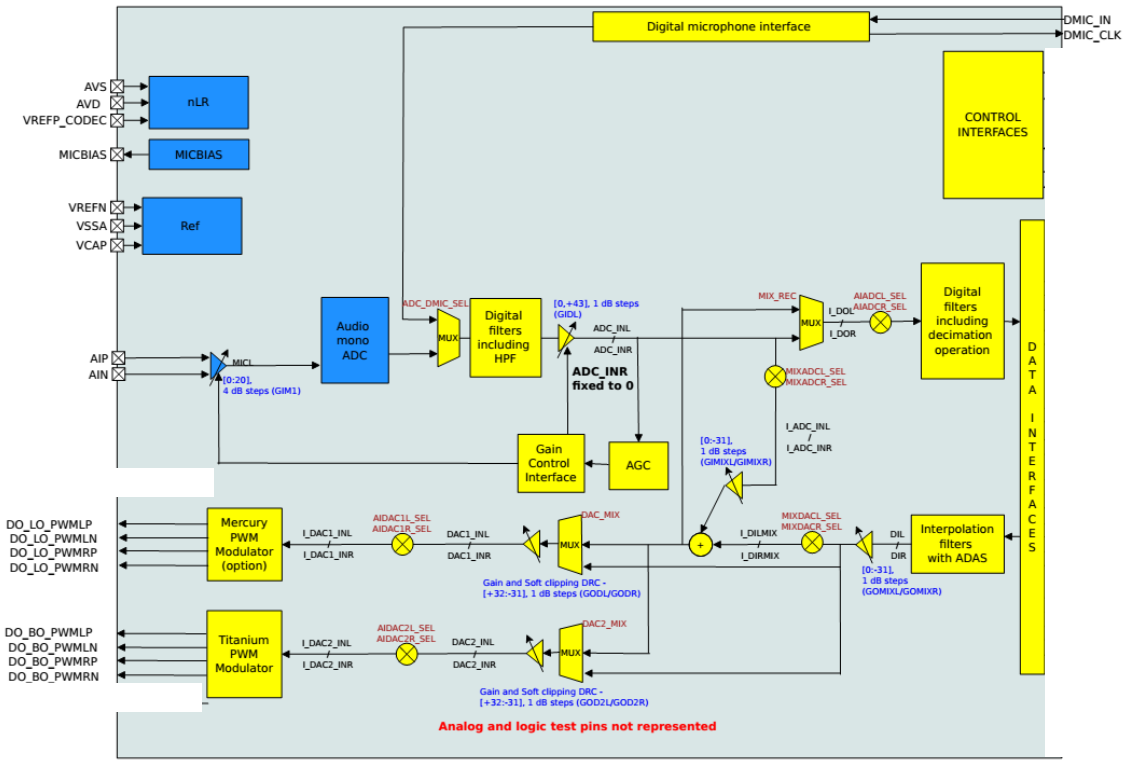


Figure 8-12 CODEC Power Diagram

There are many power parts of CODEC. Any part could be powered down independently.

8.18.1 Initial all the gain

When CODEC turn into STANDBY mode, need initial all the gain:

- Firstly set a gain that different you wished (such as '1') to GIM1 / GIM2 / GIDL / GIDR / GIL / GIR / GOL / GOR / GODL / GODR.
- Secondly wait about 10ms and than set a gain you wished (such as '0') to GIM1 / GIM2 / GIDL / GIDR / GIL / GIR / GOL / GOR / GODL / GODR.

After the step, you can change the gain you wished at any times

Power-on, power-off sequences and reset mode

Power-on sequence When the power supply ramps up, CODEC enters the power-on sequence. During this mode, the CODEC needs to be put in power-down in order to reduce audible pops.

Power-down sequence

The CODEC does not handle the power supply ramp-down by itself. The application has to turn the CODEC in power-down mode before the power supply starts its ramp-down.

8.18.2 Soft Mute mode

A soft mute mode is implemented in order to reduce audible parasites when the DAC (respectively ADC) enters or leaves the mute mode. It is controlled by the DAC_MUTE (respectively ADC_MUTE) register bit.

Setting DAC_MUTE (respectively ADC_MUTE) to '1' puts the corresponding DAC (respectively ADC) in mute mode. The CODEC decreases progressively the gain in the digital part (DWL block) from 0 dB to -∞. When the gain down sequence is completed, the output signal is equal to '0' whatever the value of the digital input data is. The CODEC then generates an IRQ and set DAC_MUTE_EVENT (respectively ADC_MUTE_EVENT) register bit to '1'.
During soft mute mode, the DAC or the ADC are still converting the data but the output final voltages are equal to VREF/2.
Conversely, when DAC_MUTE (respectively ADC_MUTE) is set to '0', the corresponding DAC (respectively ADC) leaves the mute mode by increasing progressively the gain in the digital part from -∞ to 0 dB. When the gain up sequence is completed, the DAC or the ADC returns in normal mode. The CODEC then generates an IRQ and set DAC_MUTE_EVENT (respectively ADC_MUTE_EVENT) register bit to '1'.

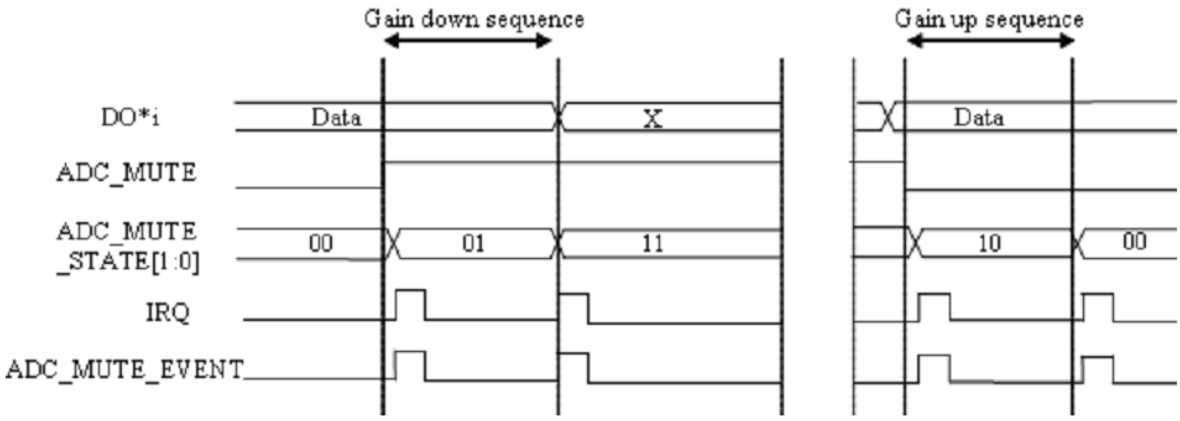


Figure 8-13 ADC Gain up and gain down sequence

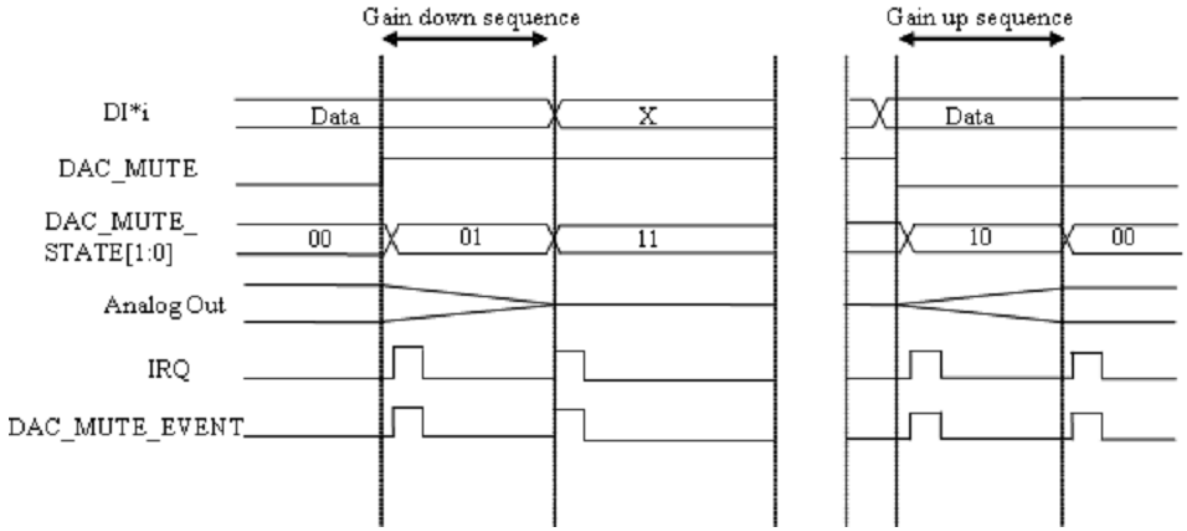


Figure 8-14 DAC Gain up and gain down sequence

The duration of gain down and gain up sequences is almost constant regardless of Fs and is about 23ms.

Do not change the value of DAC_MUTE or ADC_MUTE while the effect of the previous change is not reached (working not guaranteed).

Do not enter in power-down mode while the gain sequence is not completed (working not guaranteed).

8.18.3 Power-down and sleep modes

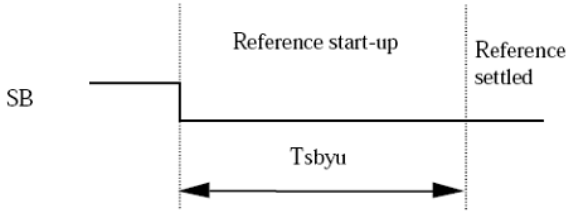
Complete power-down mode

CODEC (including all functions i.e. ADC path, and references) is put in complete powerdown mode when the CR_VIC register SB, the CR_DAC register SB_DAC, the CR_ADC register SB_ADC, AICR_DAC register SB_AICR_DAC, AICR_ADC register SB_AICR_ADC bits are set to ’1’.

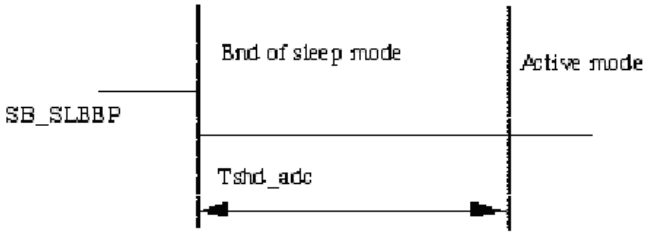
During the complete power-down mode, the power consumption is reduced to a minimum. When SB is set to ’0’, CODEC leaves the complete power-down mode. It is necessary to wait a time equal to Tsbyu before the CODEC references settles.

Sleep mode

When SB_SLEEP is set to ’1’, CODEC enters the sleep mode. Analog functions - except the voltage biasing references - enter the power-down mode. Thus, the power consumption is reduced without penalizing the start-up time.



When SB_SLEEP falls, CODEC leaves the corresponding stand-by mode; it is necessary to wait a precise time before the CODEC reaches the normal mode. Depending on the active path, this time is either called Tshd_adc for the ADC path or Tshd_dac for the DAC path.



Refer to the paragraph 3.10 for the values of Tshd_adc.

MCLK turn-off and turn-on

During the sleep mode and the complete power-down mode, the main clock MCLK may be stopped to reduce the power consumption to the leakage currents only. In all other modes, the main clock MCLK must not be stopped. The main clock MCLK must not be stopped until CODEC has reached a settled mode (complete power-down mode or sleep mode) and must be restarted before leaving this mode. When MCLK is turned off (sleep mode, complete power-down mode), writing on register values is inactive, reading the register values may not be up to date and IRQ signal is not propagated until MCLK turns on. After MCLK restarts, it is required to wait 4 MCLK cycles before reading or writing in the registers.

STANDBY PWM output pins (pop reduction)

The STANDBY_BO (respectively STANDBY_BO) output pin may be used when Titanium (respectively Mercury) sound shaper is used to drive the standby pin of an external power amplifier at PCB level. This signals allows the external amplifier to leave its standby mode once the CODEC is fully operational in order to avoid pop-up noise due to transient phenomena when leaving standby mode.

8.18.4 Working modes summary

Different working modes are sum-up in the following table (non exhaustive table):

Mode	SB	SB_SLEEP	SB_DAC2	SB_DAC	SB_AICR_ADC	SB_AICR_DAC	SB_MIC1	SB_ADC
Register signal values after a reset mode	1	1	1	1	1	1	1	1
power-down mode	1	X	1	1	1	1	X	1
Sleep mode	0	1	X	X	X	X	X	X

Record mic/line mono input mode	0	0	X	X	0	X	0	0
Playback mode, DAC to LO*outputs	X	X	X	0	X	0	X	X
Playback mode, DAC to BO*outputs	X	X	0	X	X	0	X	X

8.19 MCLK turn-off and turn-on

The main clock of CODEC is called MCLK, which is generated in CPM module and called MCLK. During the SLEEP mode and the complete power-down mode, the main clock MCLK may be stopped to reduce the power consumption to the leakage currents only. In other modes, the main clock MCLK must not be stopped.

The main clock MCLK must not be stopped until CODEC has reached the complete power-down mode and must be restarted before leaving the power-down mode.

After MCLK restarts, it is required to wait 4 MCLK cycles before reading or writing the registers.

When MCLK is turned off (SB_SLEEP=1 or SB=1), writing on register values are not taken into account, register values are not up to date when read and interrupts not generated until MCLK turns on.

8.20 Requirements on outputs and inputs selection and power-down modes

The following rules must be respected in order not to damage performances and to keep the functionality:

- If SB_MIC1 is set to 1, MICSTEREO must be equal to 0, IN_SEL, HP_SEL and LO_SEL must not be equal to ‘00’.
- If SB_MIC2 is set to 1, MICSTEREO must be equal to 0, IN_SEL, HP_SEL and LO_SEL must not be equal to ‘01’.
- If SB_LINE is set to 1, IN_SEL must not be equal to ‘10’.
- If SB_LIBY is set to 1, HP_SEL and LO_SEL must not be equal to ‘10’.
- If SB_DAC is set to 1, HP_SEL and LO_SEL must not be equal to ‘11’.

8.20.1 Initialization and configuration

To use the embedded CODEC with AIC, the following AIC registers should be set up before start the CODEC:

AICFR.ICDC = 1

AICFR.AUSEL = 1

AICFR.BCKD = 0

AICFR.SYNCD = 0

I2SCR.AMSL = 0
I2SCR.ESCLK = 1

8.21 Circuits design suggestions

This section lists a few PCB design suggestions with **different** using mode.

8.21.1 Avoid quiet ground common currents

8.21.1.1 References pins

To work properly, CODEC requires **a** few additional external components.
CODEC includes an internal voltage reference. To insure a correct common mode biasing of the internal components, an additional voltage VCAP is used. This requires connecting two decoupling capacitors (Cext) between the pin VCAP and CODEC_AVSS. One 10uF low ESR (ceramic or tantalum) and one 100nF ceramic have to be used. The ceramic capacitor has to be kept as close as possible to IC package (closer than 0.2 inch / 5 mm).

8.21.1.2 Power supply pins

CODEC drivers power supplies require external decoupling capacitors.

For each power supply, one 100 nF ceramic has to be used. The ceramic capacitor has to be kept as close as possible to IC package (closer than 0.2 inch / 5mm).

One ceramic or tantalum capacitor has to be used to decouple the analog power supply provided to the ViC. Its value has to be 10 uF.

Ideally, use separate ground planes for analog and digital parts. Connect all ground pins with thick traces to power plane in order to ensure lowest impedance connections.

All peripheral power supplies (VDDIO_CODEC and CODEC_AVD) have to be supplied by a voltage source with a noise spectrum lower or equal to the power supply noise tolerance template (PSNT2) described in the sections Power Supply Noise Tolerance Template.

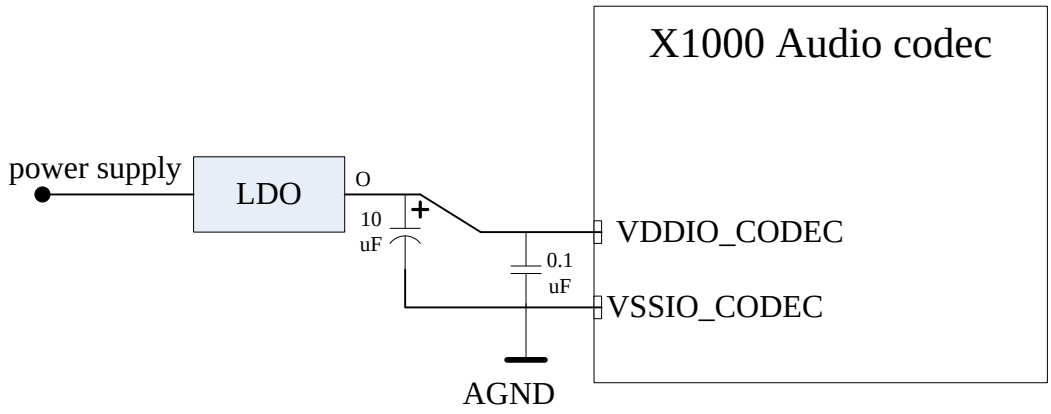


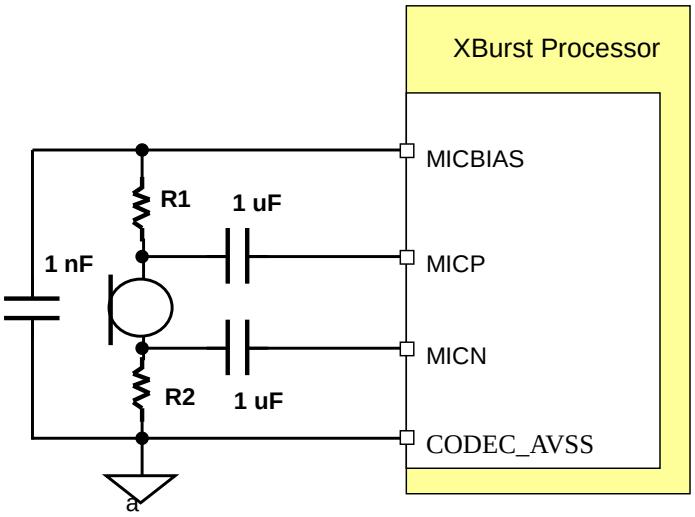
Figure 8-15 Peripheral power supply connection

8.21.2 Microphone connection

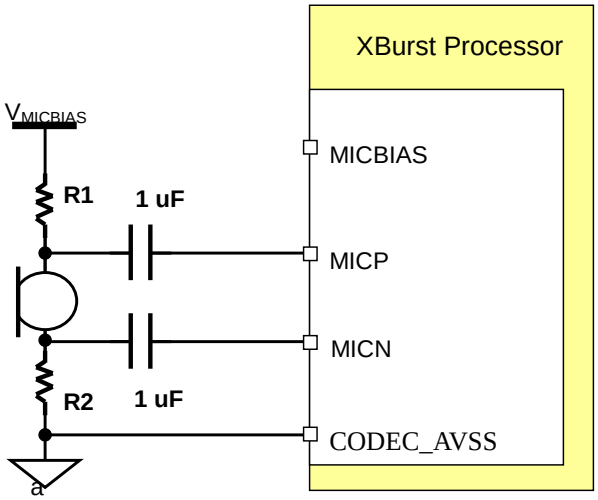
The optimal performance for the SNR is obtained in differential Microphone inputs imgingenic with a FS input level **corresponding to peak-to-peak amplitude of the signal is 0.2125V when GIM = 20 (Gain value = 20).**

We recommend customer to use differential MIC input for better performance.

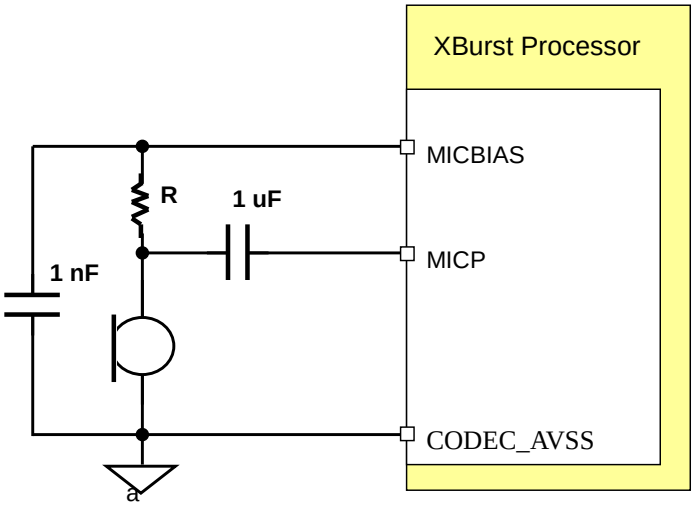
Application schematic with differential MIC input (using MICBIAS pin):



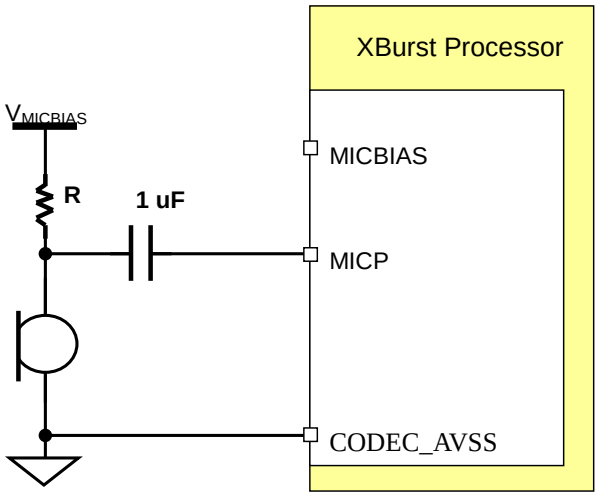
Application schematic with differential MIC input (Vmicbias generated on board):



Application schematic with single-ended MIC input (using MICBIAS pin):



Application schematic with single-ended MIC input (Vmicbias generated on board):



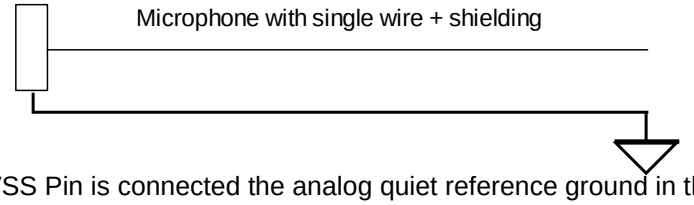
In single-ended connection, one external resistor (R) has to be used to bias the electret microphone. In differential connection, a pair of external resistor (R1, R2) has to be used to bias the electret microphone. The resistors value relation between them is R1 = R2 = R/2.

Specific value of resistor (R, commonly from 2.2k Ohm to 4.7k Ohm) and Vmicbias (if generated on board, usually from 1 to 2V or more) depends on the selected EC (Electret Condenser) microphone. The 1nf decoupling capacitance used in MICBIAS pin removes high frequency noise of the chip.

Setting SB_MIC1/SB_MIC2 to 1 will close microphone input path for saving power, also setting SB_MICBIAS to 1 will close MICBIAS stage and the MICBIAS output voltage will be zero.

MICBIAS output voltage scales with CODEC_AVDD, equals to 5/6*Fin (typical 2.08V).
MICBIAS output current is 4mA max.
MICBIAS output noise is 40uVrms max.

This configuration is better suited for microphone with single wire + shielding.



The CODEC_AVSS Pin is connected the analog quiet reference ground in the chip (refers to Grounds and analog signal references). So the ground of MIC must be connected to CODEC_AVSS using a star connection.

8.21.3 PCB considerations

Differential routing

Basic rules for differential routing:

In case of differential routing is preferred for a better noise rejection, the two differential signals must be routed in parallel on the same layer. If one signal changes for one metal layer to another, the other signal changes at the same location.

Basic spacing rules:

- spacing between the signal tracks is approximately the same as the width of the signal metal track. The spacing with other signals has to be at least 3 times the width of the metal track
- Do not route a disturbing signal in parallel to one of the differential signal: if one parasitic signal has to cross the differential signal, it has to cross the two wires.

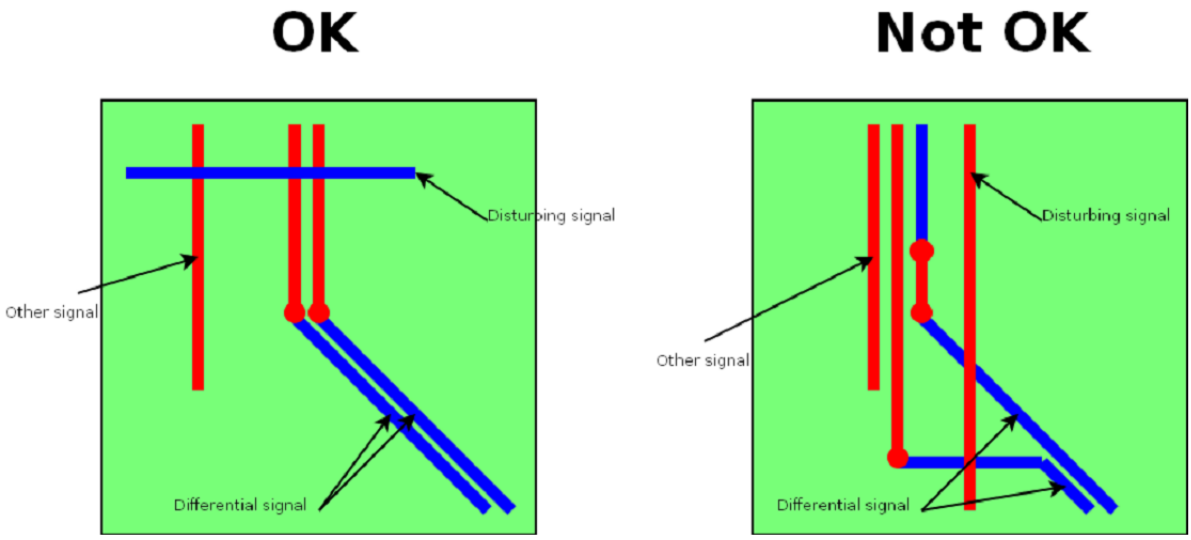


Figure 8-16 Differential routing

Routing of active filtering based application schematic

When operational amplifiers (or any power amplifier) are used in the application schematic as indicated in section ??, it is important to reduce the distance between the first set of RC filters and the output package pin of sCODa-MT1-LR.01. The track length between any signal input of the amplifier and the components must be shorter than 1 cm.

EMI consideration:

If another part of the system may be sensitive to EMI generated by the PWM switching, the first stage of the RC passive filter should be placed as close as possible from the PWM package pin of sCODa-MT1-LR.01. To simplify the components placement on PCB, this passive filter can be placed before the DC coupling capacitor as indicated on the figure below.

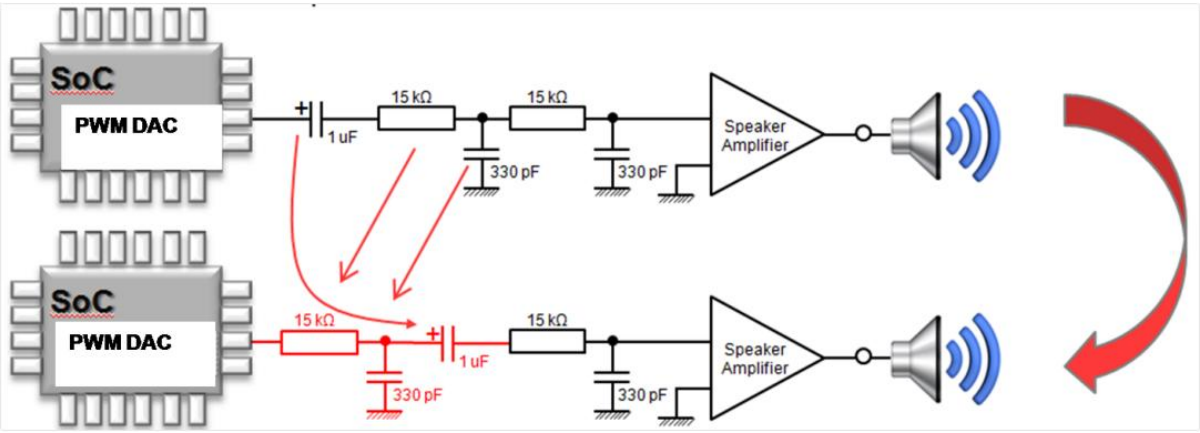


Figure 8-17 PCB optimization for avoiding EMI issue

8.21.3.1 Required external components

Component Name	Description	Value	Accuracy	Unit	DC voltage
Application schematic					
Cext1	ceramic reference decoupling capacitor on VCAP	100	+/-20%	nF	> maximum analog power supply
Cext2	tantalum or ceramic reference decoupling capacitor on VCAP	10	+/-20%	uF	
Cin	Tantalum or ceramic 3.3 V analog power supply decoupling capacitor	10	+/-20%	uF	
CV3	Ceramic AVD decoupling capacitor	100	+/-20%	nF	
CL3	Tantalum or ceramic VREFP_CODEC decoupling capacitor	10	+/-20%	uF	
CV1, CV2, CV4	Ceramic VDDA_*O_DAC decoupling capacitors	100	+/-20%	nF	
Cmic	Micbias decoupling capacitor	Refer to Microphone connection	+/-20%	nF	
Rmic	Microphone external resistor				
Cbyline	Input bypass capacitor	1	+/-20%	uF	> maximum analog power

					supply

8.22 Analog characteristics

8.22.1 Operating conditions

Typical characteristics are stated for typical process and typical operating conditions given in the following table. Minimum and maximum characteristics are stated over process variations.

Parameter	**Symbol**	**CODEC or driver pins concerned**	**Min.**	**Typ.**	**Max**	**Unit**
PCB analog ground	AGND	CODEC_AVSS, VSSIO_CODEC		0		V
Analog power supply for ADC	AVD	CODEC_AVDD	2.97	3.3	3.63	V
Analog power supply for drivers of DAC outputs	VDDA_O_DAC	VDDIO_CODEC	2.97	3.3	3.63	V
Sampling frequency	Fs		8	-	192	kHz

Parameter	**Symbol**	**CODEC or driver pins concerned**	**Min.**	**Typ.**	**Max.**	**Unit**
MCLK frequency	Fmclk	MCLK		24		MHz
MCLK duty cycle	Dmclk	MCLK	0.45	0.5	0.55	-

8.22.2 With PWM output, used for analog amplifier application

Measurement conditions:
Temperature, power supplies, Fmclk in typical conditions and Fs in the range defined in Operating conditions, unless otherwise specified.
Input sine wave with a frequency of 1 kHz, measurement bandwidth 20 Hz – 20kHz, unless otherwise specified.

Parameter	**Test conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
SNR	A-weighted, 1 kHz sine wave @ Full Scale and gains = 0 dB	105			dB
THD	1 kHz sine wave @ Full Scale -1 dB and gains = 0 dB Fs ≤ 16 kHz		-90	-85	dB
	1 kHz sine wave @ Full Scale -1 dB and gains = 0 dB Fs > 16 kHz			-80	dB
THD +N	1 kHz sine wave @ Full Scale -1 dB and gains = 0 dB Fs≤ 16 kHz		-90	-85	dB
	1 kHz sine wave @ Full Scale -1 dB and gains = 0 dB Fs > 16 kHz			-80	dB

Dynamic Range	A-weighted, 1kHz sine wave @ Full Scale -60dB and volume = 0 dB (1)	105			dB
Gain range	GODL,GODR 6-bit programmable range, digital control, @ 1 kHz	-31		+32	dB
	GOMIX 5-bit programmable range, digital control, @ 1 kHz	-31		0	dB
Gain step	GOD*, GOMIX, monotonic		1		dB
Gain accuracy	GOD*, GOMIX@ 1 kHz	-0.5		+0.5	dB
PWM frequency			800		kHz
Modulation rate		25		75	%

8.22.3 Microphone / Line input to ADC path

Measurement conditions:
Temperature, power supplies, Fmclk in typical conditions and Fs in the range defined in Operating conditions, unless otherwise specified.
Input sine wave with a frequency of 1 kHz, measurement bandwidth 20 Hz – Fs/2 for Fs = 8 to 32 kHz, measurement bandwidth 20 Hz – 20 kHz for Fs >= 44.1 kHz, unless otherwise specified.

Parameter	Test conditions	Min.	Typ	Max.	Unit
Input level (1)	Full Scale, Gain GID* = 0 dB, boost gain GIM* = 0 dB	1.89	2.12	2.39	Vpp
	Full Scale, Gain GID* = 0 dB, boost gain GIM* = 20 dB	0.189	0.212	0.239	Vpp
SNR	A-weighted, 1 kHz sine wave @ Full Scale and gain GID* = 0 dB, boost gain GIM* = 0 dB	85	90		dB
	A-weighted, 1 kHz sine wave @ Full Scale and gain GID* = 0 dB, boost gain GIM* = 20 dB	75	80		dB
THD	1 kHz sine wave @ Full Scale -1 dB and gain GID* = 0 dB, boost gain GIM* = 0 dB		-80	-70	dB
	1 kHz sine wave @ Full Scale -1 dB and gain GID* = 0 dB, boost gain GIM* = 20 dB		-70	-60	dB
Dynamic Range (2)	A-weighted, 1 kHz sine wave @ Full Scale -60 dB and gain GID* = 0 dB, boost gain GIM* = 0 dB	85	90		dB
	A-weighted, 1 kHz sine wave @ Full Scale and gain GID* = 0 dB, boost gain GIM* = 20 dB	75	80		dB
Dynamic Range over gain range (DRgr)	A-weighted, 1 kHz sine wave @ Full Scale -60 dB and gain GIL*= 0 dB, boost gain GIM* = [0-20]dB		100		dB

PSRR	100 mVpp 1 kHz sinewave is applied to AVD, input data is 0 and gain GID* = 0 dB, boost gain GIM* = 0 dB		90		dB
Gain range	Boost gain GIM* when activated	0		20	dB
	Digital gain GID*	0		+43	dB
Gain step	GIM* @1 kHz		4		dB
	GID* @1 kHz		1		dB
Gain accuracy	GIM* @1 kHz	-1		+1	dB
	GID* @1 kHz	-1		+1	dB
Input resistance (differential mic configuration)	Boost gain GIM* = 0 dB	132	160	200	kOhm
	Boost gain GIM* = 20 dB	20	26	30	kOhm
Input resistance (single-ended mic configuration)	Boost gain GIM* = 0 dB	92	115	138	kOhm
	Boost gain GIM* = 20 dB	19	24	29	kOhm

Table 8-3 Microphone/Line input performances

Measurement conditions: Temperature, power supplies, Fmclk in typical conditions and Fs in the range defined in Operating conditions, unless otherwise specified. Input sine wave with a frequency of 1 kHz, measurement bandwidth 20 Hz – Fs/2 for Fs = 8 to 32 kHz, measurement bandwidth 20 Hz – 20 kHz for Fs >= 44.1 kHz, unless otherwise specified.					
Parameter	**Test conditions**	**Min.**	**Typ**	**Max.**	**Unit**
Input capacitance	Includes 10 pF for ESD, bonding and package pins capacitances			25	pF
Input bypass capacitor	Cbyline		1		uF
Polarity	AIP-AIN to DIL		+1		

(1) Scales with VREF = VREFP_CODEC- CODEC_AVSS

(2) The specified value is extrapolated by adding 60 dB to the measured SNR

8.22.4 Micbias and reference

Measurement conditions: Temperature, power supplies, Fmclk in typical conditions and Fs in the range defined in Operating conditions, unless otherwise specified. Input sine wave with a frequency of 1 kHz, measurement bandwidth 20 Hz – 20kHz, unless otherwise specified.					
Parameter	**Test conditions**	**Min.**	**Typ**	**Max.**	**Unit**
Micbias ouput level (1)	MICBIAS_V = 0		2.08		V
	MICBIAS_V = 1		1.66		
Micbias output current				4	mA
Micbias output noise	A-Weighted		20	40	uVrms
Micbias decoupling capacitor	Cmic	0.75	1	1.25	nF
VCAP output voltage			2		V
VREFP_CODEC output level		2.35	2.5	2.65	V

8.22.5 I/O buffers

Table 8-4 I/O buffer static characteristics

Measurement conditions: Temperature and power supplies ranges defined in Operating conditions.					
Parameter	**Test conditions**	**Min.**	**Typ**	**Max.**	**Unit**
VOL	-			0.1	V
VOH	-	VDDIO_CODEC-0.1			V
IOL	-	12		24	mA
IOH	-	-24		-12	mA

Table 8-5 I/O buffer dynamic characteristics

Measurement conditions: Temperature and power supplies ranges defined in Operating conditions. Rise and fall time (10% to 90%)				
Parameter	**Min.**	**Typ**	**Max.**	**Unit**
Fall Time			5	ns
Rise Time			5	ns
Difference between Fall Time and Rise Time	0		0.2	ns
Skew between PWM outputs (*LP, *LN, *RP, *RN) (1)	0		0.5	ns

8.22.6 Characteristics of the ADC High Pass Filter

The high pass filter is a 1st order filter.

Filter characteristics	24MHz	
High Pass Filter corner frequency	-0.1 dB	11.93 Hz
	-0.5 dB	5.22 Hz
	-3 dB	1.83 Hz

8.22.7 Characteristics of the ADC Wind Noise Filter

The wind noise filter is a 1st order filter.

Filter characteristics		24MHz	
Wind Noise Filter corner frequency	Mode 1	-3 dB	59 Hz
	Mode 2	-3 dB	117 Hz
	Mode 3	-3 dB	235 Hz

Section 5

MEMORY INTERFACE

9 DDR Controller

9.1 Overview

DDRC (DDR Controller) is a general IP which provides an interface to DDR2, DDR3, mobile DDR(LPDDR) memory. The DDRC IP is designed for SOC usage and is configurable, scalable to meet the requirement of various SOC.

Features:

- Multi-port Architecture and asynchronous interface to all port
- Support clock-stop mode
- Support auto-refresh and self-refresh
- Support power-down mode and deep-power-down mode
- Programmable DDR timing parameters
- Programmable DDR row and column address width
- Programmable remapping from Logic address to Physical address

9.1.1 Supported DDR SDRAM Types

- DDR2
- DDR3
- LPDDR (mobile DDR)

Row address width less than 16-bit & Column width less than 12-bit are supported.

9.1.2 Block Diagram

Following figure shows the functional block diagram of DDR system.

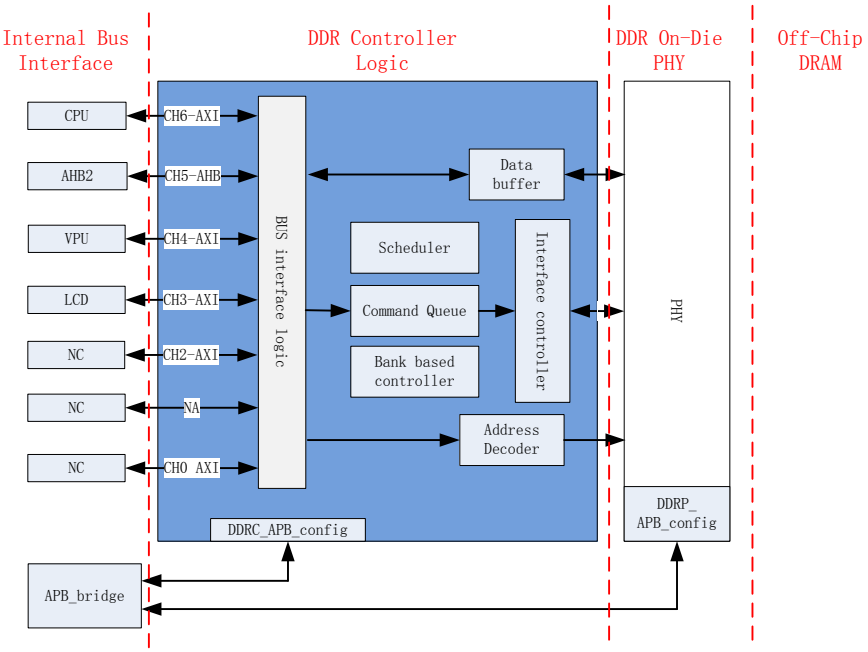


Figure 9-1 DDR system block diagram

9.2 Register Description

Table 9-1 lists the registers of DDR Controller. All of these registers are 32bit. According different configure ports to be access, these registers are divided into 3 groups.

- DDRC_AHB_CFG_GROUP:
  Registers access by AHB2(CH5_AHB)
  Base Address: *0x134F0000 ~ 0x134F0FFF*
- DDRC_APB_CFG_GROUP:
  Registers access by DDRC_APB_Config port
  Base Address: *0xB3012000 ~ 0xB3012FFF*
- DDRP_APB_CFG_GROUP:
  Registers access by DDRP_APB_Config port
  Base Address: *0xB3011000 ~ 0xB3011FFF*

Please Note that: the DDRC_AHB_CFG_GROUP must not be access under retention mode.

Table 9-1 DDRC Register

Name	Group	Address offset	Width	Access	Description
DSTATUS	DDRC_AHB	0x00	32	RW	DDR Status Register

DCFG	DDRC_AHB	0x04	32	RW	DDR Configure Register
DCTRL	DDRC_AHB	0x08	32	RW	DDR Control Register
DLMR	DDRC_AHB	0x0C	32	RW	DDR Load-Mode-Register
DTIMING 1	DDRC_AHB	0x60	32	RW	DDR Timing Configure Register 1
DTIMING 2	DDRC_AHB	0x64	32	RW	DDR Timing Configure Register 2
DTIMING 3	DDRC_AHB	0x68	32	RW	DDR Timing Configure Register 3
DTIMING 4	DDRC_AHB	0x6C	32	RW	DDR Timing Configure Register 4
DTIMING 5	DDRC_AHB	0x70	32	RW	DDR Timing Configure Register 5
DTIMING 6	DDRC_AHB	0x74	32	RW	DDR Timing Configure Register 6
DREFCN T	DDRC_AHB	0x18	32	RW	Auto-Refresh Counter
DMMAP0	DDRC_AHB	0x24	32	RW	DDR Memory CS0 Map Configure Register
DMMAP1	DDRC_AHB	0x28	32	RW	DDR Memory CS1 Map Configure Register
DDLP	DDRC_AHB	0xBC	32	RW	DFI low power handshake register
DREMAP 1	DDRC_AHB	0x9C	32	RW	DDR address remapping register1
DREMAP 2	DDRC_AHB	0xA0	32	RW	DDR address remapping register2
DREMAP 3	DDRC_AHB	0xA4	32	RW	DDR address remapping register3
DREMAP 4	DDRC_AHB	0xA8	32	RW	DDR address remapping register4
DREMAP 5	DDRC_AHB	0xAC	32	RW	DDR address remapping register5
DSTRB	DDRC_AHB	0x34	32	RW	Multi-media stride register
WCMDCT RL1	DDRC_AHB	0x100	32	RW	Write command reorder & grouping (Performance control)
RCMDCT RL0	DDRC_AHB	0x104	32	RW	Read Channel mode control (Performance control)

RCMDCTRL1	DDRC_AHB	0x108	32	RW	Read Channel mode control (Performance control)
BOUNDAEYSEL	DDRC_AHB	0x10C	32	RW	Channel boundary select
WDATTHD0	DDRC_AHB	0x114	32	RW	Wdata Channel mode control (Performance control)
WDATTHD1	DDRC_AHB	0x118	32	RW	Wdata Channel mode control (Performance control)
IPORTWPRI	DDRC_AHB	0x240	32	RW	Configuration of Internal Priority for write channel (Performance control)
IPORTRPRI	DDRC_AHB	0x244	32	RW	Configuration of Internal Priority for read channel (Performance control)
CHxWDOS	DDRC_AHB	0x200 0x204 0x208 0x20C 0x210 0x214 0x218	32	RW	WQoS configure for each channel (Performance control)
CHxRDOS	DDRC_AHB	0x220 0x224 0x228 0x22C 0x230 0x234 0x238	32	RW	RQoS configure for each channel (Performance control)
AUTOSR_CNT	DDRC_AHB	0x308	32	RW	Auto Self Refresh counter configuration
AUTOSR_EN	DDRC_AHB	0x304	32	RW	Auto Self Refresh enable
EFPRB_CFG	DDRC_AHB	0x30C	32	RW	Bus efficiency probe configuration
EFPRB_VALUE	DDRC_AHB	0x310	32	R	Bus efficiency data
CLKSTP_CFG	DDRC_APB	0x068	32	RW	Auto Clock Gating configuration
DDRC_STATUS	DDRC_APB	0x06C	32	R	DDR status
PHYRET_	DDRC_APB	0x034	32	RW	Retention Mode Configuration

CFG					
EPD_HADDR	DDRC_APB	0x070	32	RW	EPD wake up configure
EPD_LADDR	DDRC_APB	0x074	32	RW	EPD wake up configure
AUTOPRE_CFG	DDRC_APB	0x07C	32	RW	Auto Pre-charge Configuration
PHYRST_CFG	DDRC_APB	0x080	32	RW	DDR Phy reset control
CKC_CFG0	DDRC_APB	0x084	32	RW	Clock change Configuration
CKC_CFG1	DDRC_APB	0x088	32	RW	Clock change Configuration
CPM_DRCG		**0xB0000 0D0**	32	RW	CPM register, DLL & Clock control for DDR

9.2.1 DSTATUS

	DSTATUS																													0x134f0000
Bit	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
	DFI_INIT_C	Reserved																							ENDIAN	MISS	DPD	PD	AREF	SREF	CKE1	CKE0
RST	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

Bits 30~8,1: Reserved. Write has no effect, read value not need to care.
DFI_INIT_C: Read-only, indicate the DFI initialization status.

Bit [31]	Description	Remark
0	DFI initialization not completed	(reset value)
1	DFI initialization completed	

ENDIAN: Read-only, indicate the data endian status.

Bit [7]	Description	Remark
0	Little data Endian.	(reset value)
1	Big data Endian.	

MISS: Indicate the accessed address is out of memory mapping range. (This bit is un-writable. Once read this register, this bit will be set to 0)

Bit [6]	Description	Remark
0	No operation miss DDRC memory mapping.	(reset value)
1	At least one operation miss DDRC memory mapping.	

DPD: Indicate the deep-power-down status of DDR memory.

Bit [5]	Description	Remark
0	DDR memory is NOT in deep-power-down state.	(reset value)
1	DDR memory is in deep-power-down state.	

PD: Indicate the power-down status of DDR memory.

Bit [4]	Description	Remark
0	DDR memory is NOT in power-down state.	(reset value)
1	DDR memory is in power-down state.	

AREF: Indicate the auto-refresh status of DDR memory.

Bit [3]	Description	Remark
0	DDR memory is NOT in auto-refresh state.	(reset value)
1	DDR memory is in auto-refresh state.	

SREF: Indicate the self-refresh status of DDR memory.

Bit [2]	Description	Remark
0	DDR memory is NOT in self-refresh state.	(reset value)
1	DDR memory is in self-refresh state.	

CKE1: Indicate the CKE1 Pin status of DDR memory. Not support in this version.

Bit [1]	Description	Remark
0	CKE1 Pin is low.	(reset value)
1	CKE1 Pin is high.	

CKE0: Indicate the CKE0 Pin status of DDR memory.

Bit [0]	Description	Remark
0	CKE0 Pin is low.	(reset value)

1	CKE0 Pin is high.	

9.2.2 DCFG

Configure the external memory, static configuration only. ie. This register can NOT be changed on-the-fly.

	DCFG																													0x134f0004
Bit	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
	AN_TE	Reserved	ROW1			COL1			BA1	IMBA	BSL	Reserved	TYPE			ODTEN	MISPE	Reserved	ROW0			COL0			CS1EN	CS0EN	CL				BA0	DW
RST	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	0	0	0	0	0	0	0

Bits 30 , 20,14: Reserved. Writing has no effect, read as zero.

AN_TE: Analog test enable.

MISPE: Missing CS protection Enable . Set 1 to enable.

If software read (or write) a memory space which is not selected by any CS, controller will return random data to a read operation (or mask write operation) to avoid system bus being locked. A CS missing flag will set in DSTATUS.

BSL: Burst length for DDR chips

1: 8 burst

0: 4 burst

IMBA:

0: CS0, CS1 connected 2 memory chips which have same ROW, COL, BA configuration.

In this mode, both chips are configured by ROW0,COL0, BA0 . ROW1,COL1,BA1 does not effect the system.

1: CS0, CS1 connected 2 memory chips which have different ROW, COL, BA configuration. Chip0 is configured by ROW0, COL0, BA0; Chip1 is configured by ROW1, COL1, BA1;

ODTEN:

0: On-die-termination off;

1: On-die-termination on;

TYPE: Select external memory device type.

Bit [19:17]	Description	Remark
000	Reserved	(reset value)
001	Reserved	
010	Normal DDR1 (Not support in these version)	

011	Mobile DDR(LPDDR).	
100	Normal DDR2.	
101	Reserved	
110	Normal DDR3	
111	Reserved	

ROW0/1: Row Address width. Specify the row address width of external DDR.

	Description	Remark
000	12-bit row address	(reset value)
001	13-bit row address	
010	14-bit row address	
011	15-bit row address	
Reserved		

COL0/1: Column Address width. Specify the Column address width of external DDR.

	Description	Remark
000	8-bit Column address.	(reset value)
001	9-bit Column address	
010	10-bit Column address	
011	11-bit Column address	
100	12-bit Column address	
Reserved		

CS1EN: DDR Chip-Select-1 Enable.

If an DDR memory chip is connected to ddr pin cs1, set CS1EN=1.

Bit [7]	Description	Remark
0	DDR Pin CS1 is not in use.	(reset value)
1	DDR Pin CS1 is in use.	

CS0EN: DDR Chip-Select-0 Enable.

If DDR memory is connected to ddr pin cs0, set CS0EN=1.

Bit [6]	Description	Remark
0	DDR Pin CS0 is not in use.	(reset value)
1	DDR Pin CS0 is in use.	

CL: CAS Latency.

Bit [5:2]	Description	Remark
Reserved	Not in use for this version	

BA0/1: Bank Address width of DDR memory.

Bit [1]	Description	Remark
0	4 bank device, Pin ba[1:0] is valid, ba[2] un-used.	(reset value)
1	8 bank device, Pin ba[2:0] is valid.	

DW: External DDR Memory Data Width.

Specify the external DDR memory data width.

Bit [0]	Description	Remark
0	External memory data width is 16-bit.	(reset value)
1	External memory data width is 32-bit.	

9.2.3 DCTRL

On the positive edge of START, the command selected by CMD field will be performed.

	DCTRL																0x134f0008														
Bit	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
	Reserved								DFI_RST	DLL_RST	CTL_RST	CFG_RST	Reserved		KEEPSR	Reserved	ACTPD	PDT			ACTSTP	Reserved				DPD	SR	UNALIGN	ALH	SLOWEXPD	Reserved	RESET
RST	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 31~24, 19,18,16,10~7,2: Reserved. Writing has no effect, read as zero.
CFG_RST: Reserved, please keep the default value.
CTL_RST: Reserved, please keep the default value.12
DLL_RST: Reserved, please keep the default value.
DFI_RST: Reserved, please keep the default value.
KEEPSR: inner usage, keep to 0.
ACTSTP: Active Clock-Stop.

0: Clock can be stopped only after all banks be precharged

1: Clock can be stopped after some bank's row actived

ACTPD: Active Power-Down.

Some SDRAM devices support Active-Power-Down.

By default, ACTPD=0, hardware will percharge all active banks before

entering Power-Down
mode, so called Precharge-Power-Down.
By setting ACTPD=1, hardware drives SDRAM into Power-Down mode without precharge all
active banks, some banks are still active in Power-Down mode, so called Active-Power-Down.

Bit [15]	Description	Remark
0	Precharge all banks before entering power-down.	(reset value)
1	Do not precharge all banks before entering power-down.	

PDT: Power-Down Timer.
When there's no access to DDR memory for a period of time, hardware drives DDR into power-down mode to save power consumption. Hardware can exit Power-Down mode automatically when new access arrives.
If PDT=0, power-down function is disabled.
If enable power-down mode, it is recommended to be set after DDR initialization.

Bit [14:13]	Description	Remark
000	power-down disabled, hardware never drive SDRAM into power-down mode.	(reset value)
001	Enter power-down after 8 tCK idle.	
010	Enter power-down after 16 tCK idle.	
011	Enter power-down after 32 tCK idle.	
100	Enter power-down after 64 tCK idle.	
101	Enter power-down after 128 tCK idle.	
110 - 111	Reserved.	

DPD: Software drives external Mobile DDR device entering Deep-Power-Down mode.
Software set DPD = 1 drives external Mobile DDR device entering Deep-Power-Down mode instead of Power-Down mode, when there's no access to DDR memory for a period of time. So you must firstly enable Power-Down mode (refer to PDT).
Software needs to reset DDR controller and re-do a complete initial process to exit Deep-Power-Down mode.
When external device goes to Deep-Power-Down mode, it lose all data store in memory and registers.
The memory chip will cut off inner power supply for power saving.

SR: Software drive external DDR device entering Self-Refresh mode.

Software set SR=1 drive external DDR device entering self-refresh mode;

Software set SR=0 drive external DDR device exiting self-refresh mode;

In this mode, the CK to external DDR device would be stopped during self-refresh period;

But the clock supplied to ddr_controller logic would not stop.

Software can read & write ddr_controller registers in this mode.

Software can NOT read or write memory data in this mode.

NOTE: Software must guarantee that there's no memory access during self-refresh mode. Otherwise, software can NOT exit this mode, **system would be hang-up!!**

Bit [5]	Description	Remark
0	Drive external DDR device entering self-refresh mode.	(reset value)
1	Drive external DDR device exiting self-refresh mode.	

UNALIGN: Enable unaligned transfer on AXI BUS. (Not useful in this version)

Bit [4]	Description	Remark
Reserved	Not use in this version	

ALH: Advanced Latency Hiding.

This is a test-oriented register.

Some latency is reduced in special cases.(Not useful in this version)

Bit [3]	Description	Remark
0	Disable ALH.	(reset value)
1	Enable ALH.	

SLOWEXPD: Slow exit power-down.

This register select how to entry power-down mode. DLL is off in slow exit power-down mode.

Bit [3]	Description	Remark
0	Slow exit power-down disenable.	(reset value)
1	Slow exit power-down enable.	

CKE: Control the status of CKE pin.

Write CKE=1 can set CKE pin to HIGH state.

Write CKE=0 would be ignored.

The default value of CKE Pin is low;
CKE0,1 Pins status is represented by DDR_STATUS register.
Caution: This register is used only for DDR initializing sequence; software should NOT update this register when DDR memory is in normal working mode.

Bit [1]	Description	Remark
0	Not set CKE Pin High.	(reset value)
1	Set CKE Pin to HIGH.	

RESET: Module reset for ddr_controller.
Software is able to reset ddr_controller by setting RESET bit high and ends the reset by clear this bit.

Bit [0]	Description	Remark
0	End resetting ddr_controller.	(reset value)
1	Resetting ddr_controller.	

9.2.4 DLMR

DLMR register is used for initializing the DDR SDRAM memory device.
On the positive edge of START, the command selected by CMD field will be performed.

	DLMR																0x134f000C													
Bit	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
	DDR_ADDR																Reserved		BA		Reserved		CMD		Reserved		START					
RST	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 7~6, 2~1: Reserved. Writing has no effect, read as zero.
DDR_ADDR: When performing a DDR command, DDR_ADDR[13:0] corresponding to
external DDR address Pin A[13:0]; DDR_ADDR[15:14] are reserved.

Bit [31:12]	Description	Remark
0000_0000	{MA[9:0],OP[9:0]} for LPDDR2(not support in this version) For other DDR the low bit 14 bit corresponding to external DDR address Pin A[13:0].	(reset value)

BA: Bank Address.

When performing a DDR command, BA[2:0] corresponding to external DDR address Pin BA[2:0].

Bit [10:8]	Description	Remark
000	corresponding to external DDR address Pin BA[2:0].	(reset value)

CMD: Select command to process when setting START from low to high.

On the positive edge of START, one of the following commands will be performed.

Bit [5:3]	Description	Remark
000	Precharge one bank / All banks. (dependent field : BA, DDR_ADDR)	(reset value)
001	Auto-Refresh.	
010	Load Mode Register. (dependent field : BA, DDR_ADDR)	
010	ZQCS for DDR3	
100	ZQCL for DDR3	
101~111	Reserved.	

START: Start to perform a command to external DDR memory.

The command is performed on the positive edge of START; Hardware will clear START bit to zero when command issued out to external DDR memory.

Write 0 to START will be ignored and take no effect;

START=1 means hardware is busy executing current command and can NOT accept new command;

Software must check START=0 before writing 1 to START.

Bit [0]	Description	Remark
0	No command is performed.	(reset value)
1	On the positive edge of START, perform a command defined by CMD field.	

9.2.5 DTIMING1,2,3,4,5,6 (DDR Timing Configure Register)

The timing parameters are identical to the JEDEC DDR Specification.

	DTIMMING1																														0x134f0060	
Bit	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

	Reserved							tRTP								Reserved								tWTR								Reserved								tWR								Reserved								tWL							
RST	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

	DTIMMING2																															0x134f0064																			
Bit	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																						
	Reserved							tCCD								Reserved								tRAS								Reserved								tRCD								Reserved								tRL							
RST	0	0	0	0	0	0	2	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1																						

	DTIMMING3																															0x134f0068																			
Bit	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																						
	Reserved		ONUM			tCKSRE			Reserved				tRP				Reserved				tRRD				Reserved				tRC																												
RST	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1																						

	DTIMMING4																															0x134f006C																			
Bit	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																						
	Reserved			tRFC					tEXTRW		tRWCOV		tCKE				Reserved					tMINSR			Reserved		tXP			Reserved		tMRD																										
RST	0	0	0	0	1	1	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	1																						

	DTIMMING5																															0x134f0070																			
Bit	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																						
	tCTLUPD								Reserved				tRTW				Reserved				tRDLAT				Reserved				tWDLAT																												
RST	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1																						

	DTIMMING6																															0x134f0074																			
Bit	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																						
	tXSRD								Reserved				tFAW				Reserved				tCFGW				Reserved				tCFGR																												
RST	1	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1																						

If the DDR clock period are tCK.

All these registers's value equal to the number tCK cycle. For example, if tRTP=2, it means 2 tCK between READ and PRECHARGE;

tRTP: READ to PRECHARGE command period.
tWTR: WRITE to READ command delay.
tWR: WRITE Recovery Time defined by register MR of DDR2 DDR3 memory.
tWL: Write latency, please notice that this version only support AL=0.
tCCD: CAS# to CAS# command delay
tRAS: ACTIVE to PRECHARGE command period.
tRAS defines the ACTIVE to PRECHARGE command period to the same bank.
tRCD: ACTIVE to READ or WRITE command period.
tRCD defines the ACTIVE to READ/WRITE command period to the same bank.
tRL: Read latency, please notice that this version only support AL=0.
ONUM: Keep to 4 in this version.
tRP: PRECHARGE command period
tRP defines the PRECHARGE to next command period to the same bank.
tRRD: ACTIVE bank A to ACTIVE bank B command period.
tRRD defines the ACTIVE to ACTIVE command period to **different** banks.
tRC: ACTIVE to ACTIVE command period.
tRC defines the ACTIVE to ACTIVE command period to the same bank.
tWDLAT: tWL-1(when use LPDDR2, set tWDLAT=tWL).
tRDLAT: tRL-2. (when use LPDDR2, set tRDLAT=tRL).
tFAW: 4-active command window.
tCFGW: Write PHY configure registers to other commands delay. Not need to change.
tCFGR: Ready PHY configure registers to other commands delay. Not need to change.
tRTW: Read to Write latency.
tCTLUPD: Inner usage. Not need to change.
tEXTRW: keep the default value.
tRWCOV: keep the default value.

All these registers' value has different rate between the tCK cycle
tXSRD: exit self-refresh to READ delay.
Delay time is tXSRD*4 (tCK)
Note: You can set this registers as tXS(or tXSNR), but must make sure the system do not read ddr before tXSRD in wakeup.

tCKSRE: Valid clock after enter self-refresh(tCKSRX = tCKSRE in this version).
Delay time is tCKSRE*8 (tCK)

tCKE: minimum CKE pulse width.

tCKE define the minimum CKE pulse width, include high level and low level.

Bit [18:16]	Description	Remark
000	1 tCK.	(reset value)
001	2 tCK.	
010	3 tCK.	
011	4 tCK.	
100	5 tCK.	
101	6 tCK.	
110	7 tCK.	
111	8 tCK.	

tRFC: AUTO-REFRESH command period.

tRFC defines the minimum delay after an AUTO-REFRESH command. During tRFC period, no command can be issued to DDR memory.

Delay Time = **2 * (tRFC + 1).**

Bit [29:24]	Description	Remark
000000	1 tCK.	(reset value)
000001	3 tCK.	
000010	5 tCK.	
000011	7 tCK.	
… …	**… 2 * (tRFC + 1) …**	
111101	125 tCK.	
111110	127 tCK.	
111111	129 tCK.	

* tCK – one DDR memory clock cycle, typical tCK value is 7.5 ns (133MHz clock).

tMINSR: Minimum Self-Refresh / Deep-Power-Down time.

After DDR memory turns into Self-Refresh or Deep-Power-Down mode, it will NOT exit until tMINSR condition meets.

Delay Time = tMINSR * 8 + 1.

Bit [11:8]	Description	Remark
0000	1*8 + 1 tCK.	(reset value)
0001	2*8 + 1 tCK.	
0010	3*8 + 1 tCK.	
0011	4*8 + 1 tCK.	
… …	**… tMINSR * 8 + 1 …**	
1101	14*8 + 1 tCK.	

1110	15*8 + 1 tCK.	
1111	16*8 + 1 tCK.	

tXP: EXIT-POWER-DOWN to next valid command period.

tXP defines the EXIT-POWER-DOWN to next valid command period to all banks.

Bit [6:4]	Description	Remark
000	1 tCK.	(reset value)
001	1 tCK.	
010	2 tCK.	
011	3 tCK.	
100	4 tCK.	
101	5 tCK.	
110	6 tCK.	
111	7 tCK.	

tMRD: Load-Mode-Register to next valid command period.

tMRD defines the Load-Mode-Register to next valid command period.

Bit [1:0]	Description	Remark
00	1 tCK.	(reset value)
01	2 tCK.	
10	3 tCK.	
11	4 tCK.	

9.2.6 DREFCNT (DDR Auto-Refresh Counter)

	DREFCNT																														0x134f0018
Bit	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
	Reserved								CON								CNT								Reserved				CLK_DIV			REF_EN
RST	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits 31~24, 7-4: Reserved. Writing has no effect, read as zero.

CON: A constant value used to compare with the CNT value.

After reset, CON=0xFF and CNT=0x00;

It is not recommended to set CON=0x00.

CNT: 8-bit counter; When the value of CNT match the value of CON, flag bit EQU is set high

and an auto-refresh command will be issued to DDR memory. READ only.

CLK_DIV : Clock Divider.

Divide the dclk to generate a lower frequency of clock to drive the auto-refresh counter. This helps to save power consumption.
Set CLK_DIV=0 can disable the clock to auto-refresh counter.

When the DDR memory is in self-refresh mode or in deep-power-down mode, disable the clock of auto-refresh counter can save power consumption. Future more, the module clock to DDRC can also be stopped.

dclk is CKO clock, When ddr work in 500Mbps, dclk is 250Mhz.

Bit [3:1]	Description	Remark
000	dclk / 16.	(reset value)
001	dclk / 32.	
010	dclk / 64.	
011	dclk / 128.	
100	dclk / 256.	
101	dclk / 512.	
110	dclk / 1024.	
111	--	

REF_EN: Enable Refresh Counter.
Software set REF_EN=1 right after initialize ddr memory.

Bit [29]	Description	Remark
0	Enable auto-refresh counter.	(reset value)
1	Disable auto-refresh counter.	

9.2.7 DMMAP0,1 (DDR Memory Map Configure Register)

The physical base address and size of external DDR Memory can be configured by DMMAP register.
The size of external DDR Memory must be: **2^(24+n), n=0, 1, 2, 3, ….**

When the following equation is met:
(AXI_BUS_Address[31:24] & MASK[7:0]) == BASE
The DDR Memory is selected.

	DMMAP0,1																0x134f0024, 0x134f0028															
Bit	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
	Reserved																BASE								MASK							

RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bits 31~16: Reserved. Writing has no effect, read as zero.
BASE: base address.
MASK: address mask.
Examples:

1 DDR address space in system memory : 0x2000_0000 ~ 0x2FFF_FFFF (256MB)
BASE=0x20 MASK=0xF0.

2 DDR address space in system memory : 0x5000_0000 ~ 0x57FF_FFFF (128MB)
BASE=0x50 MASK=0xF8.

NOTE: If DDRC is disabled, please set DMMAP=0x0000_FF00 (reset value).

9.2.8 DDLP (DDR DFI low power handshake control register)

	DDLP																														0x134f00BC	
Bit	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
	Reserved													Only	UAR	UZQ	SLP				FLP				Reserved					FSR	FPD	LPEN
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1

Bits 31~19: Reserved. Writing has no effect, read as zero.
Bits 18~16: Inner usage, keep to 0
LPEN: DFI low power(lp) interface enable
FPD: 1: Use fast lp handshake process during power down.
0: Use slow lp handshake process during power down.
FSR: 1: Use fast lp handshake process during self-refresh.
0: Use slow lp handshake process during self-refresh.
SLP, FLP: slow LP handshake and fast LP handshake timing register, Just keep it as reset value.

9.2.9 DREMAP1,2,3,4,5 (DDR Address Remapping Register 1,2,3,4,5)

	DREMAP1																														0x134f009C	
Bit	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
	Reserved			BIT3MP					Reserved			BIT2MP					Reserved			BIT1MP				Reserved			BIT0MP					
RST	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

	DREMAP2																0x134f00A0															
Bit	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
	Reserved			BIT7MP					Reserved			BIT6MP					Reserved			BIT5MP				Reserved			BIT4MP				
RST	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0

	DREMAP3																0x134f00A4															
Bit	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
	Reserved			BIT11MP					Reserved			BIT10MP					Reserved			BIT9MP				Reserved			BIT8MP				
RST	0	0	0	0	1	0	1	1	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0

	DREMAP4																0x134f00A8															
Bit	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
	Reserved			BIT15MP					Reserved			BIT14MP					Reserved			BIT13MP				Reserved			BIT12MP				
RST	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0

	DREMAP5																0x134f00AC															
Bit	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
	Reserved			BIT19MP					Reserved			BIT18MP					Reserved			BIT17MP				Reserved			BIT16MP				
RST	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0

DREMAP1~5 are used to define address mapping in DDRC. The “BITnMP” represents address[31:20]. The low 4KB(bit [11:0]) address are reserved without any change. The high 20 bits are configurable. For example, If you want to remap address between address[27:25] and address[14:12], you can set BIT12MP as 0, BIT13MP as 1, BIT14MP as 2, BIT0MP as 12, BIT1MP as 13 and BIT2MP as 14.

Original address:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11~0
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Reserved

Remapped address

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11~0

19	18	17	16	2	1	0	12	11	10	9	8	7	6	5	4	3	15	14	13	Reserved

9.2.10 WCMDCTRL1 (Performance wcmd reorder & grouping)

Terminology:

- Priority: priority of command, the command with higher priority will be processed prior to that with lower priority. The priority is generated from:
  1. on-port control signals (hpri, ARPRI, AWPRI)
  2. iport_pri (register configurable for each port)
  3. QoS counter
- Page-hit-conflit:

  Page-hit: wpage-hit

  Page-conflict: the page of write command is conflict with the page of read command. In this case, to guarantee the read-write coherency, the write command must be process prior to read command.

	WCMDCTRL1																												0x134f0100	
Bit	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
	wcmd_lock_thd					wpage_hit_max								pend_qos_cnt												wram_in_use					disable_reorder	msk_empty_wfifo
RST	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bits	Name	Description	Reset
[31:27]	wcmd_lock_thd	Indicates how many wcmd(write commands) are grouped together. Higher value improves total bandwidth of ddr, but leads to longer read command latency.	0x8
[26:19]	wpage_hit_max	Wpage-hit affects the reorder of wcmd, the wcmd with same page should be bonded for higher performance. This register indicates the maximum number of wcmd to be bonded. Higher value improves total bandwidth of ddr, but leads to longer read maximum latency.	0x8
[18:7]	pend_qos_cnt	Once the wcmd is accepted, a counter indicates how long the command is pending, when the pending time is exceed Pend_QoS once, the priority of this command will increase 1, up to 3, which increases the priority to be processed.	0x100

		This mechanism avoids the wcmd with lower priority pending too long. Range: 0x001 ~0xFFF **NOTE**: configure step: 1. set d*isable_reorder* to 0. 2. execute *step1* for 14 times 3. set wcmd_pend_qos to expect value 4. set disable_reorder to 1, (Optional) Higher value improves total bandwidth of ddr, but leads to longer maximum latency for write command with lower priority. **Dynamic configuration is not supported.**	
[6:2]	wram_in_use	0~12. Ram to be used in write command reorder. Reduction of this number leads to poor performance. **Configuration is not recommended**	0xb
[1]	disable_reorder	0: wcmd will be reordered based on priority and page-hit-conflict 1: wcmd will not be reordered, the commands will be process with the order of being received. **Dynamic configuration is not recommended.**	0
[0]	msk_empty_wfifo	0: wcmd is involved in arbitration immediately. 1: wcmd is not involved in arbitration until receives enough wdata.	0

9.2.11 RCMDCTRL0 (Performance rcmd request control)

Terminology:

rfifo_thd: read fifo threshold. Read command is not involved in arbitration until the vacancy in rfifo is higher than rfifo threshold. This should improves the ddr performance when processing accessing discrete pages. However, for page-access (commands access the same page continuously) the performance will be lagged as the threshold affects the command grouping.

rcmd_igr_cflit: guarantee coherency between wcmd and rcmd. If the coherency if not required in some application (such as LCD), ignoring of confliction would improve ddr read performance.

	RCMDCTRL0																															0x134f0104
Bit	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

	ch3_rcmd_igr_cflit	ch3_rfifo_thd_en	ch3_rfifo_thd						Reserved								ch1_rcmd_igr_cflit	ch1_rfifo_thd_en	ch1_rfifo_thd						Reserved							
RST	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0

Bits	Name	Description	Reset
[31]	ch3_rcmd_igr_cflit	1: enable. Consistence between wcmd and rcmd will not be guaranteed. 0: disable.	0x0
[30]	ch3_fifo_thd_en	0: disable rfifo_thd (recommanded) 1: enable rfifo_thd	0x0
[29:24]	ch3_rfifo_thd	If Ch3 rfifo level threshold higher than this value, rcmd request is masked	0x8
[23]	reserved		0x0
[22]	reserved		0x0
[21:16]	reserved		0x8
[15]	ch1_rcmd_igr_cflit	1: enable. Consistence between wcmd and rcmd will not be guaranteed. 0: disable.	0x0
[14]	ch1_fifo_thd_en	0: disable rfifo_thd (recommanded) 1: enable rfifo_thd	0x1
[13:8]	ch1_rfifo_thd	If Ch3 rfifo level threshold higher than this value, rcmd request is masked	0x8
[7]	reserved		0x0
[6]	reserved		0x1
[5:0]	reserved		0x8

9.2.12 RCMDCTRL1 (Performance rcmd request control)

	RCMDCTRL1																															0x134f0108
Bit	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
																	Ch5_rcmd_igr_cflit	Ch5_rfifo_thd_en	Ch5_rfifo_thd						Ch4_rcmd_igr_cflit	Ch4_rfifo_thd_en	Ch4_rfifo_thd					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0

Bits	Name	Description	Reset
[15]	ch5_rcmd_igr_cflit	0: disable rfifo_thd (recommended) 1: enable rfifo_thd	0x0
[14]	ch5_fifo_thd_en	1: enable. consistence between wcmd and rcmd will not be guaranteed. 0: disable.	0x1
[13:8]	ch5_rfifo_thd	If Ch1 rfifo level threshold higher than this value, rcmd request is masked	0x8
[7]	ch4_rcmd_igr_cflit	0: disable rfifo_thd (recommended) 1: enable rfifo_thd	0x0
[6]	ch4_fifo_thd_en	1: enable. consistence between wcmd and rcmd will not be guaranteed. 0: disable.	0x0
[5:0]	ch4_rfifo_thd	If Ch1 rfifo level threshold higher than this value, rcmd request is masked	0x8

9.2.13 BOUNDARYSEL (Channel boundary select)

	BOUNDARYSEL																														**0x134f010C**
Bit	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
	Reserved	CH6_block_wready	CH5_block_wready	CH4_block_wready	CH3_block_wready	CH2_block_wready	CH1_block_wready	CH0_block_wready	Reserved								CH7_boundary		CH6_boundary		CH5_boundary		CH4_boundary		CH3_boundary		CH2_boundary		CH1_boundary		CH0_boundary	
RST	0	1	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

Bits	Name	Description	Reset
[30]	CH6_block_wready	0: No wready block (default) 1: Wready bolck Debug usage, don't change this bit	0x1
[29]	CH5_block_wready	0: No wready block (default) 1: Wready bolck Debug usage, don't change this bit	0x0
[28]	CH4_block_wready	0: No wready block (default) 1: Wready bolck Debug usage, don't change this bit	0x0
[27]	CH3_block_wready	0: No wready block (default) 1: Wready bolck Debug usage, don't change this bit	0x0
[26]	CH2_block_wready	0: No wready block (default)	0x0

		1: Wready bolck Debug usage, don't change this bit	
[25]	CH1_block_wready	0: No wready block (default) 1: Wready bolck Debug usage, don't change this bit	0x0
[24]	CH0_block_wready	0: No wready block (default) 1: Wready bolck Debug usage, don't change this bit	0x0
[15:14]	CH7_boundary	0: 512byte boundary (default) 1: 1024byte boundary 2: 2048byte boundary 3: 4096byte boundary	0x0
[13:12]	CH6_boundary	0: 512byte boundary (default) 1: 1024byte boundary 2: 2048byte boundary 3: 4096byte boundary	0x0
[11:10]	CH5_boundary	0: 512byte boundary (default) 1: 1024byte boundary 2: 2048byte boundary 3: 4096byte boundary	0x0
[9:8]	CH4_boundary	0: 512byte boundary (default) 1: 1024byte boundary 2: 2048byte boundary 3: 4096byte boundary	0x0
[7:6]	CH3_boundary	0: 512byte boundary (default) 1: 1024byte boundary 2: 2048byte boundary 3: 4096byte boundary	0x0
[5:4]	CH2_boundary	0: 512byte boundary (default) 1: 1024byte boundary 2: 2048byte boundary 3: 4096byte boundary	0x0
[3:2]	CH1_boundary	0: 512byte boundary (default) 1: 1024byte boundary 2: 2048byte boundary 3: 4096byte boundary	0x0
[1:0]	CH0_boundary	0: 512byte boundary (default) 1: 1024byte boundary 2: 2048byte boundary 3: 4096byte boundary	0x0

9.2.14 WDATTHD0 (performance wcmd request control)

Terminology:

Wfifo_thd: wdata fifo threshold. In axi port, as the wdata might be far behind of wcmd. The threshold guarantees the wcmd will not be involved in arbitration until receive enough wdata. 11

	WDATTHD0																													0x134f0114
Bit	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
	Reserved	Ch3_wfifo_thd_en	Ch3_wfifo_thd						Reserved								Ch1_wfifo_thd_en	Ch1_wfifo_thd							Reserved								
RST	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	0	1	0	0	0

Bits	Name	Description	RW
30	Ch3_wfifo_thd_en	1: Enable 1: Disable (recommended)	0x0
[29:24]	Ch3_wfifo_thd	Wcmd is involved in arbitration before receiving all wdata.	0x8
[22]	Ch2_wfifo_thd_en	1: Enable 1: Disable (recommended)	0x0
[21:16]	Ch2_wfifo_thd	Wcmd is involved in arbitration before receiving all wdata.	0x8
[14]	Ch1_wfifo_thd_en	1: Enable 1: Disable (recommended)	0x0
[13:8]	Ch1_wfifo_thd	Wcmd is involved in arbitration before receiving all wdata.	0x8
[6]	Reserved		0x0
[5:0]	Reserved		0x8

9.2.15 WDATTHD1 (performance wcmd request control)

	WDATTHD1																													0x134f0118
Bit	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
	Reserved																Ch5_wfifo_thd_en	Ch5_wfifo_thd							Reserved	Ch4_wfifo_thd_en	Ch4_wfifo_thd					
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0

Bits	Name	Description	RW
[14]	Ch5_wfifo_thd_en	1: Enable 1: Disable (recommended)	0x1

[13:8]	Ch5_wfifo_thd	Wcmd is involved in arbitration before receiving all wdata.	0x8
[6]	Ch4_wfifo_thd_en	1: Enable 1: Disable (recommended)	0x0
[5:0]	Ch4_wfifo_thd	Wcmd is involved in arbitration before receiving all wdata.	0x8

9.2.16 IPORTPRI (performance priority control)

Terminology:

Priority affects on arbitration between ports. Transfers with higher priority will be grant first

Port_pri: priority provided by masters.

Iport_pri: register configurable which applies to each ports. The finial priority is the larger value of port_pri and iport_pri.

	IPORTWPRI																												0x134f0240	
Bit	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
	Reserved								Ch7_iport_wpri_en	Ch6_iport_wpri_en	Ch5_iport_wpri_en	Ch4_iport_wpri_en	Ch3_iport_wpri_en	Ch2_iport_wpri_en	Ch1_iport_wpri_en	Ch0_iport_wpri_en	Ch7_iport_wpri		Ch6_iport_wpri		Ch5_iport_wpri		Ch4_iport_wpri		Ch3_iport_wpri		Ch2_iport_wpri		Ch1_iport_wpri		Ch0_iport_wpri	
RST	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
	IPORTRPRI																												0x134f0244	
Bit	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
	Reserved								Ch7_iport_rpri_en	Ch6_iport_rptr_en	Ch5_iport_rptr_en	Ch4_iport_rptr_en	Ch3_iport_rptr_en	Ch2_iport_rptr_en	Ch1_iport_rptr_en	Ch0_iport_rptr_en	Ch7_iport_rptr		Ch6_iport_rptr		Ch5_iport_rptr		Ch4_iport_rptr		Ch3_iport_rptr		Ch2_iport_rptr		Ch1_iport_rptr		Ch0_iport_rptr	
RST	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

Bits	Name	Description	RW
[23:16]	Chn_iport_wpri_en Chn_iport_rpri_en	Enable iport_pri (iport_wpri is write channel priority, Iport rpri is read channel priority) 0: disable, use external priority 1: enable, use the max priority of external and internal priority	0x0
[15:0]	Chn_iport_wpri Chn_iport_rpri	Priority of channel 0x3: the highest priority ...	0x0

		0x0: the lowest priority	

9.2.17 CHxQOS0,1,2,3,4,5 (performance QoS control)

Terminology:

Qos: Generally speaking, the Qos controller is a timer, it indicates the how long the command is pending. Once the timer exceed predefined threshold (con), the priority increases by 1, up to top-limit (max). The higher priority is, the sooner it will be processed. This register affects the latency.

	CHxWQOS0,1,2,3,4,5,6																0x134f0200,204,208,20C,210,214,218													
Bit	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
	Qos_en	Outstanding_en	Reserved												max		con															
RST	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	CHxRQOS0,1,2,3,4,5,6																0x134f0300,304,308,30C,310,314,318													
Bit	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
	Qos_en	Outstanding_en	Reserved												max		con															
RST	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	Name	Description	Rrset
[31]	Qos_en	enable	0x1
[17:16]	Max	Up-limit of priority	0x3
[15:0]	Con	Threshold of timer	0x100

9.2.18 AUTOSR_CNT

Terminology:

AUTOSR: auto self-refresh. When the DDRC bus interface is idle for some time, the DDRC can issue a command to put the DRAM into self-refresh mode. This will same much power in off-chip DRAM.

	AUTOSR_CNT																0xB3010308													
Bit	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

	Reserved	BUS_IDLE_CNT
RST	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

Bits	Name	Description	Rrset
[19:0]	BUS_IDLE_CNT	Timer Cnt: when the bus interface is idle(all transfer are finished), the internal timer is going to run, after the Timer Cnt is meet, DDR issues Self-Refresh command.	0x100

9.2.19 AUTOSR_EN

	AUTOSR_EN																0xb3010304													
Bit	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
	Reserved																															AUTOSR_EN
RST	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

Bits	Name	Description	Rrset
[0:0]	BUS_IDLE_EN	0: disable auto self-refresh 1: enable auto self-refresh	0x0

9.2.20 CLKSTP_CFG

Description:

Under certain condition, the DDRC input clock can be stopped for power saving. For example, when the SDRAM is in self-refresh mode, the clock can be stop.

	CLKSTP_CFG																0xB3012068													
Bit	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
	clkstp_en	idle_cond_en	lp_cond_en	sr_cond_en	Reserved																dly_timer											
RST	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	0	0	0	0	0

Bits	Name	Description	Rrset
[31]	clkstp_en	Function enable	

[30]	idle_cond_en	Enable judge condition: ddrc idle	
[29]	lp_cond_en	Enable judge condition: low-power mode (DCTRL.PDT)	0x0
[28]	sr_cond_en	Enable judge condition: self-refresh	0x0
[11:0]	dly_timer	Specify the time before shutting down the clock after all condition met.	0x20

Please Note that:
The CLKSTP will affect the function of lp(low power mode), auto-sr(auto self-refresh)mode. If the Clock is gating, the timer used for lp/auto-sr will stop running. As a result, if the Auto-self-refresh enabled, the bit [28] should be set.

9.2.21 DDRC_STATUS

Description:
Read-Only register indicates the ddrc status. This register can be accessed even in DDR retention mode.

DDRC_STATUS																																0xB301206C
Bit	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
	Reserved													ckc_reject	ckc_ack	ckc_req	Reserved		ddr_sr_mode	ddr_lp_mode	ddrc_idle	ch6_bus_idle	ch5_bus_idle	ch4_bus_idle	ch3_bus_idle	ch2_bus_idle	ch1_bus_idle	ch0_bus_idle	Reserved			clkstp_req
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	Rrset
[13]	ddr_sr_mode	0: DRAM is not in self-refresh mode 1: DRAM is in self-refresh mode	0
[12]	ddr_lp_mode	0: ddr PHY is not in low-power mode 1: ddr PHY is in low-power mode	0
[11]	ddrc_idle	Indicates the ddrc control logic is idle	1
[10:4]	ch*x*_bus_idle	Indicates the bus interface for each channel is idle	1
[3:1]	Reserved		0
[0]	clkstp_req	Indicates the ddrc is requiring CPM gating ddrc clock	0

9.2.22 PHYRET_CFG

Description:
DDR PHY can be set into retention mode, which allow power cutting-down. This can save the leakage power in highest level.

However, when use this function, please flow the retention flow.

	PHYRET_CFG																															0xB3012034
Bit	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
	Reserved																			cfg_iso_en	Reserved			phy_iso_en	Reserved			clk_iso_en	Reserved		ret_en_n	ret_en
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0

Bits	Name	Description	Rrset
[12	cfg_iso_en	PHY output prdata isolation enable	0
[8]	phy_iso_en	PHY output dfi signal isolation enable (high-active)	0
[4]	clk_iso_en	PHY output clock isolation enable	0
[1]	ret_en_n	Retention enable: low-active	1
[0]	ret_en	Retention enable: High-active (make sure: ret_en = ~ret_en_n)	0

9.2.23 PHYRST_CFG

Soft Reset of DDRC

	REG_NAME																															0xB3012080
Bit	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
	Reserved							dfi_reset	dll_reset	ctl_reset	cfg_reset	Reserved																				phy_reset
RST	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

Bits	Name	Description	Rrset
[24]	dfi_reset	Soft reset of dfi	0
[23]	dll_reset	Soft reset of dll	0
[22]	ctl_reset	Soft reset of ctl	0
[21]	cfg_reset	Soft reset of cfg	0
[0]	phy_reset	Soft reset of PHY	0

9.2.24 CPM_DRCG

	CPM_DRCG																															0xB00000D0
Bit	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

	Reserved	DLLRESET	CFGCLKEN
RST	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

Bits	Name	Description	Rrset
[31:2]	Reserved	Inner usage, keep 0x10	0x0
[1]	DLLRESET	1: reset the DLL in DDR PHY 0: disable the reset.	0x0
[0]	CFGCLKEN	Inner usage, keep to 1	0x1

9.3 Functional Description

9.3.1 DDRC and DDR2 Memory Initialization Sequence

The following content is just an example for initialization, the sequence is already implementation in our BSP.

Example

One 512Mb x16 DDR2 device connected on CS0;
No memory device connected on CS1;
DCK = 133 MHz, CL = 3.

1 After system reset, wait system clock stable before initialize DDRC.
2 Configure the Clock-Control module for DDRC clocks.
3 DDR Memory device need at least 200us initialization time after power-on before it can accept any command.

//----------
// INIT DDRC
//----------
4 Configure DCFG = 0x.
5 Configure DTIMING1~6 = 0x.
6 Configure DMMAP0 = 0x.
7 Configure DMMAP1 = 0x0000FF00.
//----------
// INIT DDR PHY
//----------
8 Configure DDR PHY and finish PHY training process.(relate to DDRPHY spec)
//----------
// DDRC performance control configure (optional)
//----------

9 Configure RCMDCTRL0 = 0x08080808 (optional)
10 Configure RCMDCTRL1 = 0x00000808 (optional)
11 Configure WDATTHD0 = 0x08080808 (optional)
12 Configure WDATTHD1 = 0x00000808 (optional)
//----------
// INIT DDR memory device
//----------
13 Set CKE Pin HIGH : Configure DCTRL = 0x00000002.
//----------
// Enable Refresh Counter
//----------
14 Enable Refresh Counter : Configure DREFCNT = 0x.
15 AUTO-REFRESH : Configure DCTRL = 0x.
//----------
// END INITIALIZING SEQUENCE
//----------

9.4 Change Clock Frequency

To reduce power consumption, the system clock frequency may be changed frequently according to the application. There are 3 ways to change the clock frequency.

9.4.1 Manually SELF-REFRESH Mode

DDR can stay in SELF-REFRESH & DEEP-POWER-DOWN mode for a long period of time. System clock frequency can be changed during this time. Even more, the clocks to DDRC module can also be stopped to save power-consumption.
Reference Sequence:

1 Manually issue SELF-REFRESH command to DDR.
2 Change relative register in CPM.
3 Change system clock frequency.
4 Drive DDR exit SELF-REFRESH mode.

9.4.2 CPM driven SELF-REFRESH Mode

CPM will auto drive DDRC to self-refresh mode, when use ddr2, ddr3.
To change clock frequency, please refer CPM spec.

9.4.3 DLL bypass mode

The clock frequency on which DLL in ddr phy works much faster than 200MHz, As the result, when ddr clock is failed to meet this condition (for example, work in 12MHz) , the DLL must be bypassed in

order to supply clock to ddr controller.

Step1, Configure DLL into reset mode to guarantee a driving clock.

(0xB00000D0) CPM_DRCG[1] <= 1;

Step2, Configure DLL into bypass mode.

(0xB3011014) DDR_ACDLLCR[0] <= 1;

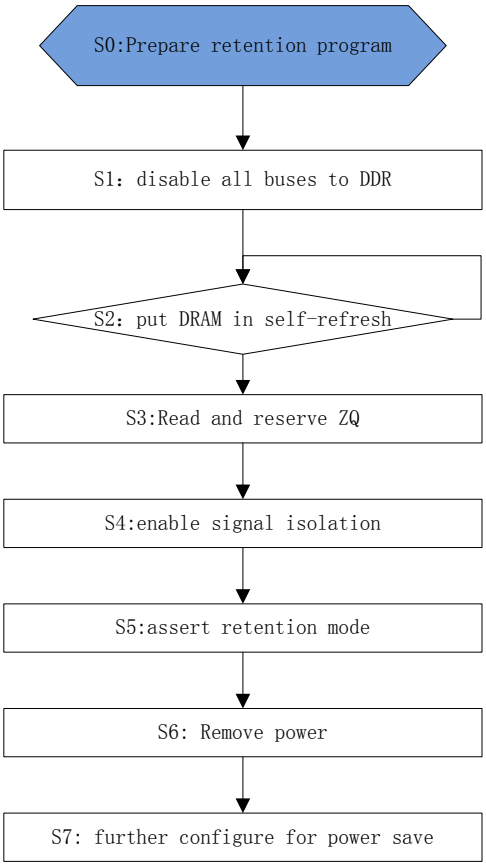
Then, (0xB00000D0) CPM_DRCG[1] <= 0;

9.5 Data Endian

Fix to little Endian.

9.6 Retention Flow

9.6.1 Enter Retention mode



Description:

S0: Prepare Retention Program

Before perform retention enter flow, the program for this flow should be mapped to on chip RAM

(such as TCSM in PDMA), during retention mode, any content in DRAM is not accessible. The PC of CPU should jump out of DDR range to on chip RAM

S1: Disable All Bus

Disable all device which addressing DDR, only DDRC APB/ DDRP APB port can be active.

S2: Put DRAM in self-refresh mode

there are two ways to drive DRAM into self-refresh

1. manually configuration (please refer DCTRL.SR)
2. Auto-self-refresh (AUTOSR_EN) (*recommended*)

polling the DDRC_STATUS.ddr_sr_mode to make sure the DRAM is in self-refresh mode

S3: Read ZQ value and reserve it future use.

(refer: X1000_ddrphy_spec: ZQnSR0.ZCTRL)

S4: Enable signal isolation

PHYRET_CFG.cfg_iso_en <= 1

PHYRET_CFG.clk_iso_en <= 1

S5: Assert DDR PHY Retention

PHYRET_CFG.ret_en <= 1

PHYRET_CFG.ret_en_n <= 0

S6: Remove Power

please refer Power off sequence

S7: Further Operation for power saving

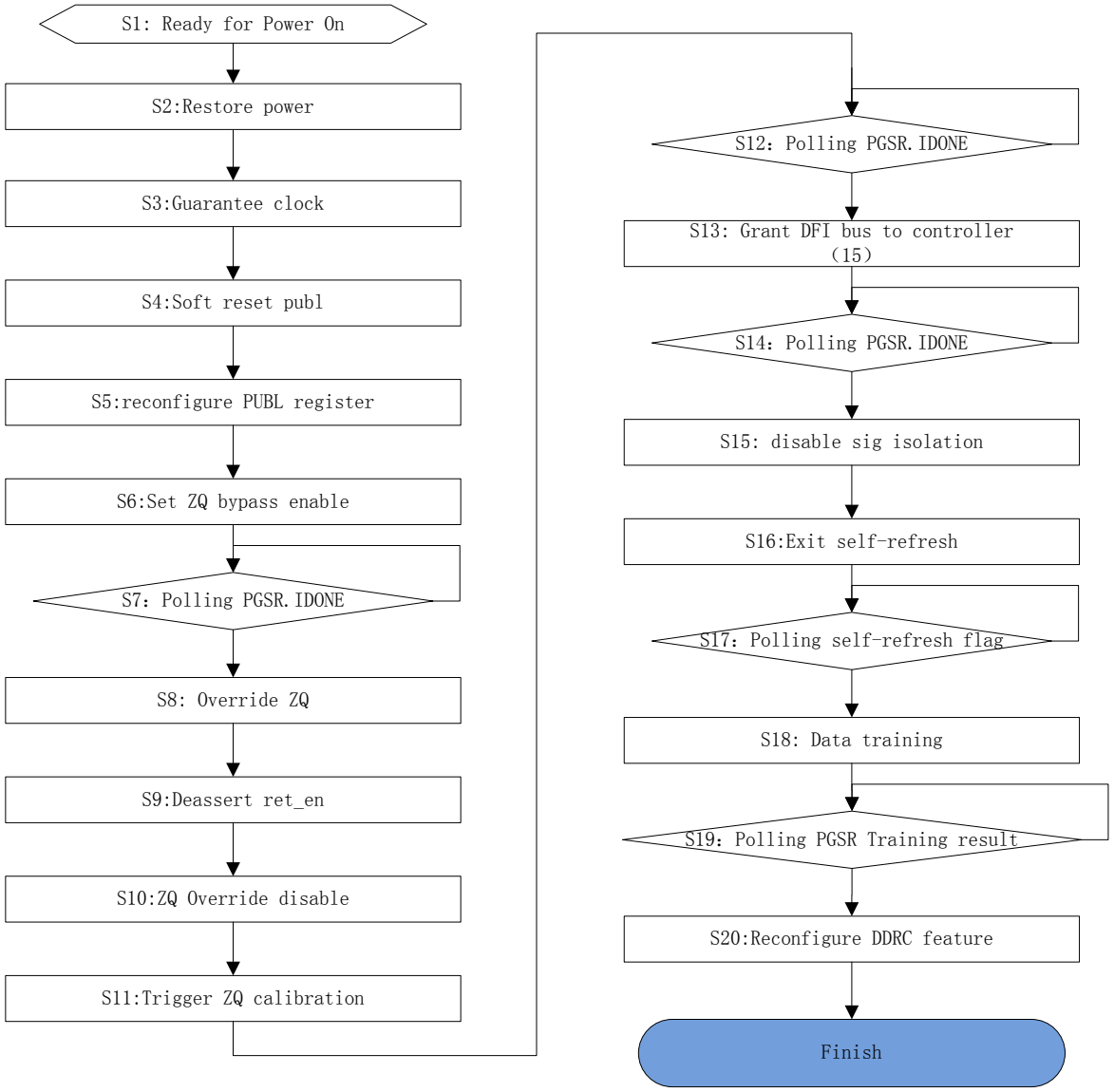
- Configure CPM, gating DDR clock
- Others

Now, the DDR is safely in retention mode, any transfer to DDR from CH0~CH6 is forbidden. Please refer 'Exit Retention mode' for wake DDR

Please note that, the retention mode is only used in system sleep mode, which still powering on GPIO. Because GPIO is must for exiting retention operation.

9.6.2 Exit Retention mode

Soft ware control flow



Description:

S1: Ready for Power-On

S2: Restore DDR Power

Please refer Sequence of DDR Power up.

S3: Guarantee clock

- Turn off CPM: ddr gating if it is gated in '*Enter Retention Mode: S7*'.
- Disable PHY clock isolation
  PHYRET_CFG.clk_iso_en <= 0
- Disable Auto-clk-stp mode
  CLKSTP.clkstp_en <= 0

S4: Soft reset DDR PHY

1) PHYRST_CFT.*_reset <= 1
2) Delay for some cycles
3) PHYRST_CFT.*_reset <= 0

S5: Reconfigure PUBL registers

Please refer '*X1000_ddrphy_spec*'

S6: Set ZQ bypass enable

DDR_PHY_PIR.ZCLBYP <= 1

S7: Polling register

DDR_PHY_PGSR.IDONE = 1

S8: Override ZQ

DDR_PHY_ZQnCR0.ZDEN <= 1 + DDR_PHY_ZQnCR0.ZDATA <= previous value (in '*Enter Retention Mode: S3*').

S9: De-assert ret_en

PHYRET_CFG.ret_en <= 0 + PHYRET_CFG.ret_en_n <= 1

S10: Override ZQ disable

DDR_PHY_ZQnCR0.ZDEN <= 0

S11: Trigger ZQ Calibration

DDR_PHY_PIR.ZQCAL <= 1 + DDR_PHY_PIR.INIT <= 1

S12: Polling register

DDR_PHY_PGSR.IDONE = 1

S13: Grant DFI bus to DDRC

DDR_PHY_PIR.CTLDINIT <= 1 + DDR_PHY_PIR.INIT <= 1

S14: Polling register

DDR_PHY_PGSR.IDONE = 1

S15: Disable signal isolation

PHYRET_CFG.cfg_iso_en <= 0

S16: Exit self-refresh

- If system works in Auto-self-refresh mode, disable this function
  AUTOSR_EN <= 0
- if works in manually self-refresh mode. Configure register to exit
  DCTL.SR <= 0

S17: Polling register to ensure ddrc exit self-refresh mode

DDRC_STATUS.ddr_sr_mode = 0

S18: Perform Data training

Please note that it will override some data in DDR, so make sure the training address is not in use.

DDR_PHY_PIR.DQTRN <= 1 + DDR_PHY_PIR.DRAMINIT <= 1

S19: Polling register

DDR_PHY_PGSR.DTDONE = 1 + DDR_PHY_PGSR.DTERR = 0

S20: Reconfigure DDRC feature:

eg:

- AUTODR_EN <= 1
- CLKSTP_CFG.clkstp_en <= 1
- ..... others .....

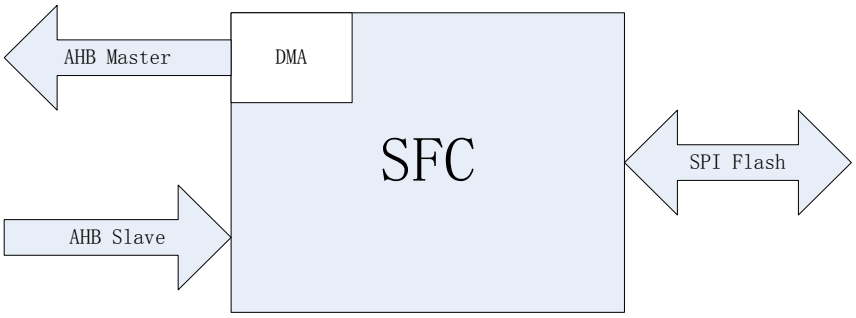
S21: Finish

DDR is thoroughly get out of Retention mode. Change PC to DDR for normal operation.

10 SPI Flash Controller(SFC)

10.1 Overview

This module is an SPI master to control the SPI flash device. Data can be transferred in CPU controlling mode or DMA mode.



10.2 Features

- SPI protocol support: Standard, Dual, Quad SPI
- transmit-only or receive-only operation
- MSB always be first in intra transfer of one byte. Least Significant Byte first for inter transfer of data bytes, and Most Significant Byte first for inter transfer of command or address bytes.
- 64 entries x 32 bits wide data FIFO
- one device select
- Configurable sampling point for reception
- Configurable timing parameters: tSLCH, tCHSH and tSHSL
- Configurable flash address wide are supported
- 7 transfer formats: Standard SPI, Dual-Output/Dual-Input SPI, Quad-Output/Quad-Input SPI, Dual-I/O SPI, Quad-I/O SPI, Full Dual-I/O SPI, Full Quad-I/O SPI
- two data transfer mode: slave mode and DMA mode
- Configurable 6 phases for software flow

10.3 Block Diagram

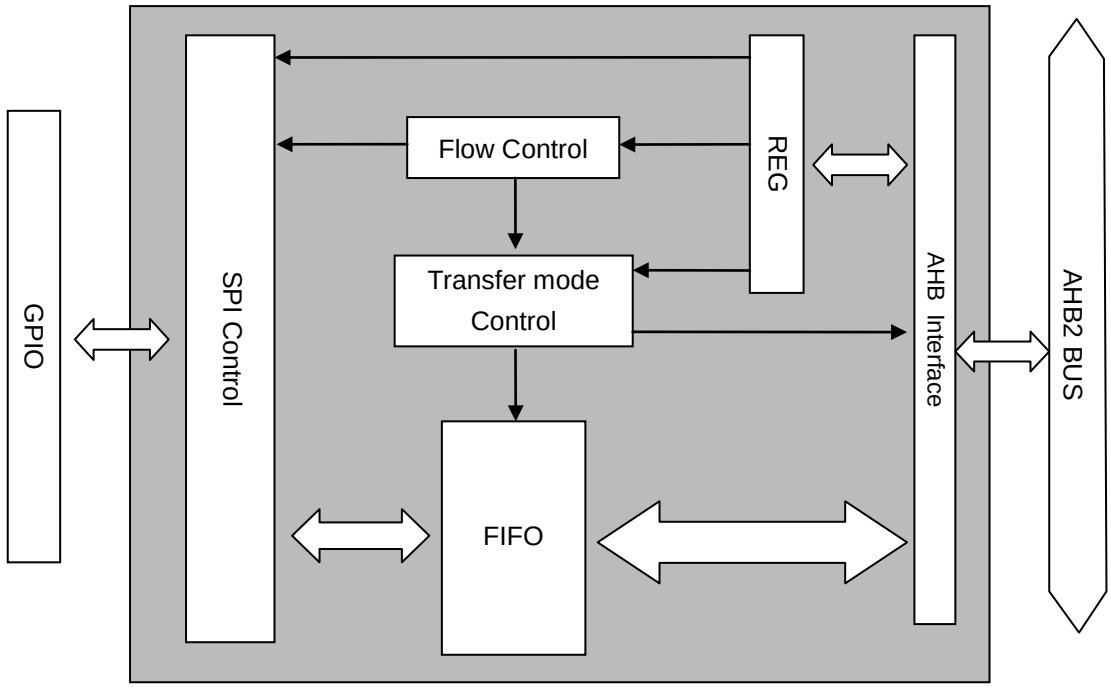


Figure 10-1 SFC Block Diagram

The AHB Interface include one slave interface and one master interface. The slave interface is used for register configuration and slave transfer mode, and the master interface is designed for DMA operation.

The FIFO block is bidirectional, used as data buffer for transmission(from AHB to device) and reception(from device to AHB).

The Flow Control module controls the operation of the 6 phases

The Transfer mode Control act as the handler of slave mode and DMA mode.

10.4 Functional Description

10.4.1 Bus function

10.4.1.1 AHB Slave Interface

- parameter configuration
- transfer data in slave mode.
- data and address must align with 4 bytes.
- only SINGLE transfer is supported.

10.4.1.2 AHB Master Interface

- transfer data in DMA mode
- SINGLE, INCR, INCR4, INCR8, INCR16, INCR32 are supported.

10.4.2 Configurable Timing parameter

10.4.2.1 Propagation delay

The propagation delay do not effect the data transmission, and it would not effect the data reception when the serial clock has a low frequency, as shown in the Figure 10-2

But when the serial clock run at a fast frequency, the propagation delay will cause the incoming data could not arrival at the right clock edge, then the sampling point should be adjustable as shown in the Figure 10-3

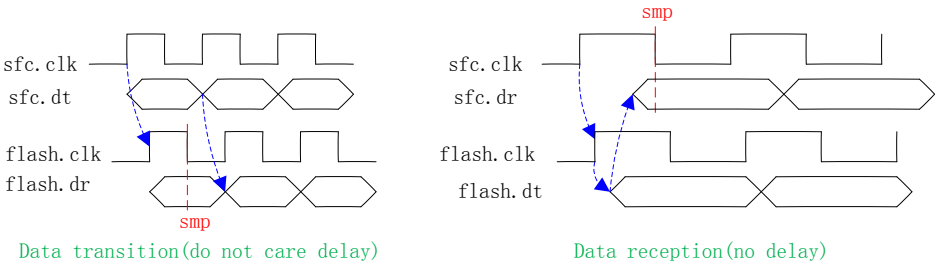


Figure 10-2 Sample point(no delay)

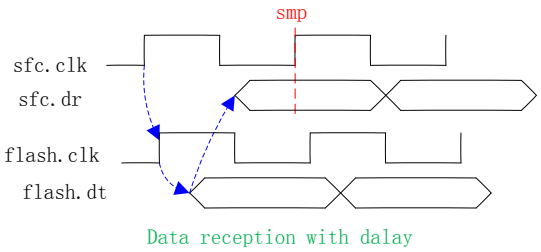


Figure 10-3 Sample point(has delay)

10.4.2.2 AC timing

tSETUP(tSLCH), tHOLD(tCHSH), tSH(tSHSL) are configurable.

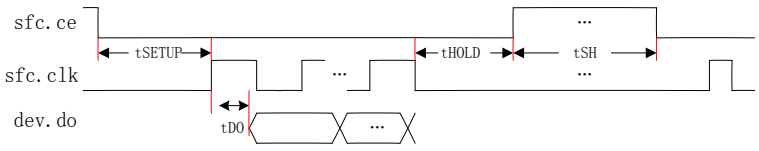


Figure 10-4 AC timing

10.5 Pin Description

Table 10-1 I/O Pin Description

Name	I/O	Description
SFC_CLK	Output	Serial clock
SFC_CE	Output	Device select
SFC_DT_IO0	InOut	Serial data IO0
SFC_DR_IO1	InOut	Serial data IO1
SFC_WP_IO2	InOut	Serial data IO2(WP#)
SFC_HOLD_IO3	InOut	Serial data IO3(HOLD#)

10.6 Data Format Description

10.6.1 Transfer format

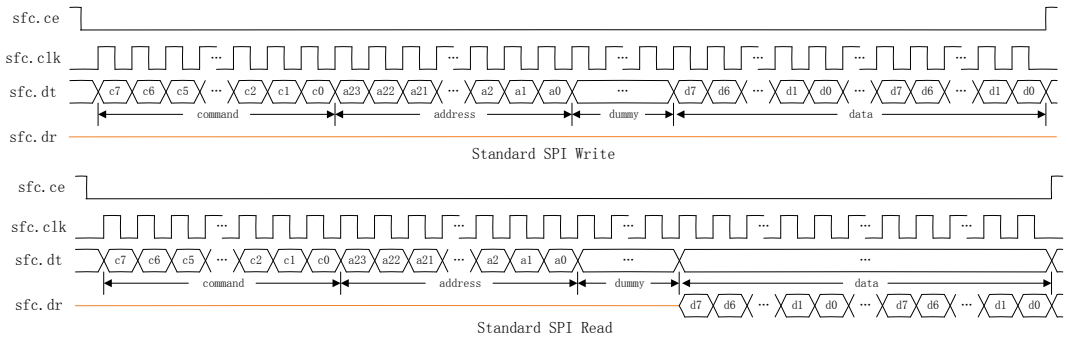


Figure 10-5 Data Format(Standard SPI)

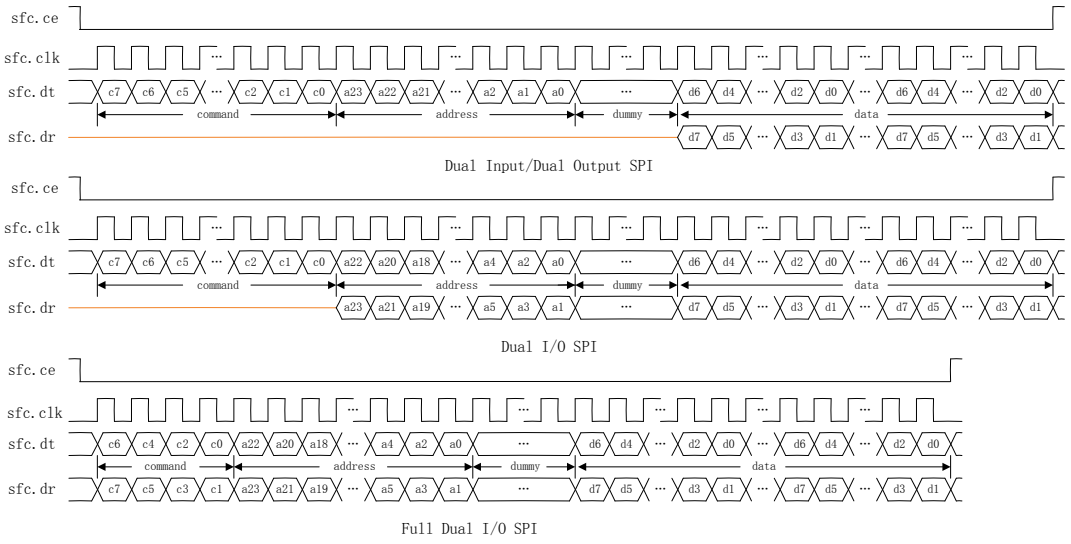


Figure 10-6 Data Format(Dual SPI)

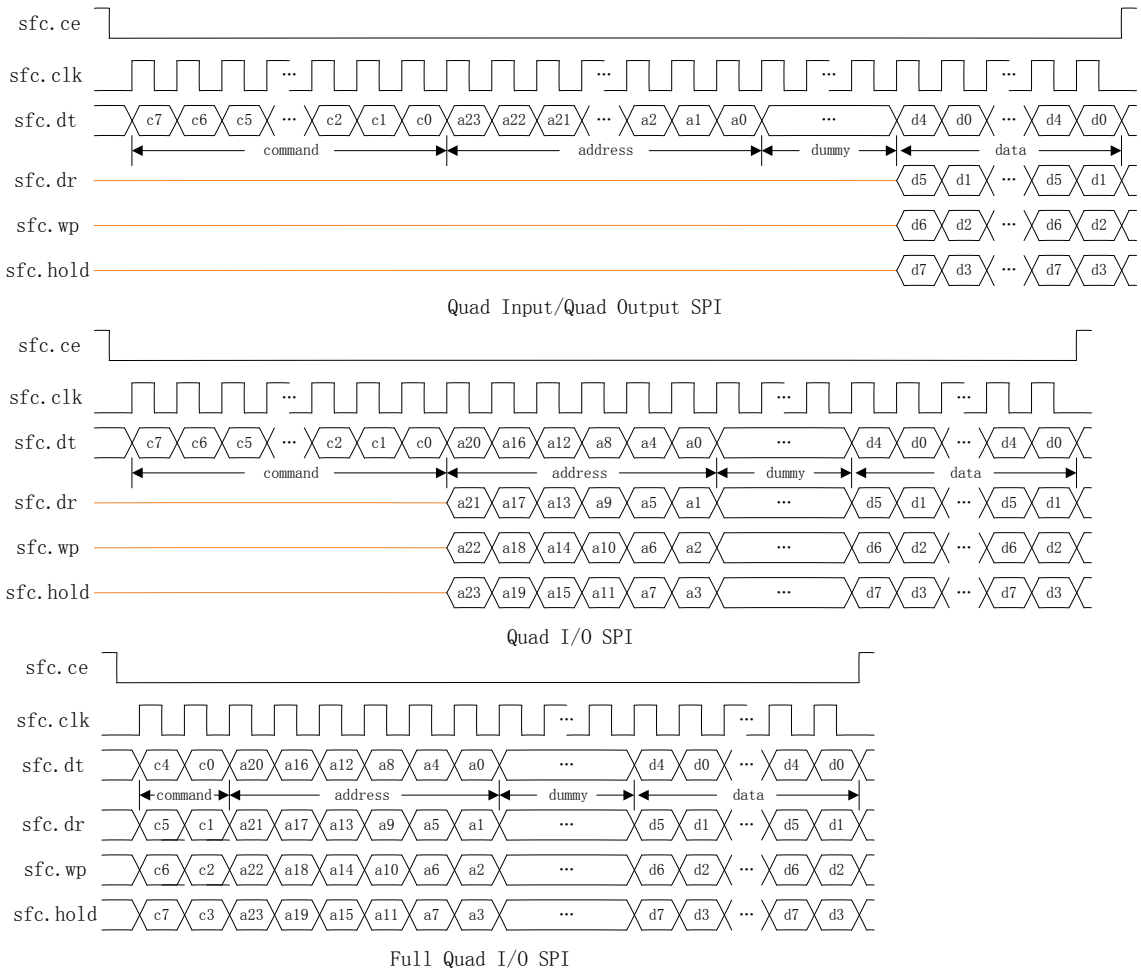


Figure 10-7 Data Format(Quad SPI)

10.6.2 Endian Description

Command and data are MSB first, and address is LSB first, as shown in the Figure 10-5, Figure 10-6 and Figure 10-7

10.7 Registers Description

10.7.1 Instructions

Following chapter will descript the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. The registers can be read and written by AHB2 bus
3. Register read/write attribute

- ✧ R - Read only
- ✧ W - Write only
- ✧ RW - read and write
- ✧ RCW - read and write, but clear to 0 by read
- ✧ RSW - read and write, but set to 1 by read
- ✧ RWC - read and write, clear to 0 by write 1, write 0 has no effect
- ✧ RWS - read and write, set to 1 by write 1, write 0 has no effect
- ✧ RC - read only, and clear to 0 by read
- ✧ RS - read only, and set to 1 by read
- ✧ SPEC - special access method, relate to its description

4. Reset Value
   - 1 - reset to 1
   - 0 - reset to 0
   - ? - value unknown after reset

10.7.2 Map

The SFC's registers map base address is 0x13440000

The SFC is controlled by a set of registers that the application configures before every operation. The following table lists all the registers.

Table 10-2 SFC Registers Map

Name	Reset Value	Address Offset	Description	RW	Mask
GLB	0x0000_008F	Base+0x0000	Global register	RW	0x0000_2FFF
DEV_CONF	0x0000_0027	Base+0x0004	Device parameter	RW	0x0002_FFFF
DEV_STA_EXP	0x0000_0000	Base+0x0008	Expected status of device	RW	0xFFFF_FFFF
DEV_STA_RT	0x0000_0000	Base+0x000c	Real-time status of device	R	0xFFFF_FFFF
DEV_STA_MSK	0x0000_0000	Base+0x0010	Mask flag of device's status	RW	0xFFFF_FFFF
TRAN_CONF0	0x0000_0000	Base+0x0014	Transfer parameter of phase0	RW	0xFFFF_FFFF
TRAN_CONF1	0x0000_0000	Base+0x0018	Transfer parameter of phase1	RW	0xFFFF_FFFF
TRAN_CONF2	0x0000_0000	Base+0x001c	Transfer parameter of phase2	RW	0xFFFF_FFFF
TRAN_CONF3	0x0000_0000	Base+0x0020	Transfer parameter of phase3	RW	0xFFFF_FFFF
TRAN_CONF4	0x0000_0000	Base+0x0024	Transfer parameter of phase4	RW	0xFFFF_FFFF
TRAN_CONF5	0x0000_0000	Base+0x0028	Transfer parameter of phase5	RW	0xFFFF_FFFF
TRAN_LEN	0x0000_0000	Base+0x002c	Transfer length	RW	0xFFFF_FFFF
DEV_ADR0	0x0000_0000	Base+0x0030	Device address of phase0	RW	0xFFFF_FFFF
DEV_ADR1	0x0000_0000	Base+0x0034	Device address of phase1	RW	0xFFFF_FFFF
DEV_ADR2	0x0000_0000	Base+0x0038	Device address of phase2	RW	0xFFFF_FFFF
DEV_ADR3	0x0000_0000	Base+0x003c	Device address of phase3	RW	0xFFFF_FFFF
DEV_ADR4	0x0000_0000	Base+0x0040	Device address of phase4	RW	0xFFFF_FFFF
DEV_ADR5	0x0000_0000	Base+0x0044	Device address of phase5	RW	0xFFFF_FFFF

DEV_PLUS0	0x0000_0000	Base+0x0048	Plus address of phase0	RW	0xFFFF_FFFF
DEV_PLUS1	0x0000_0000	Base+0x004c	Plus address of phase1	RW	0xFFFF_FFFF
DEV_PLUS2	0x0000_0000	Base+0x0050	Plus address of phase2	RW	0xFFFF_FFFF
DEV_PLUS3	0x0000_0000	Base+0x0054	Plus address of phase3	RW	0xFFFF_FFFF
DEV_PLUS4	0x0000_0000	Base+0x0058	Plus address of phase4	RW	0xFFFF_FFFF
DEV_PLUS5	0x0000_0000	Base+0x005c	Plus address of phase5	RW	0xFFFF_FFFF
MEM_ADDR	0x0000_0000	Base+0x0060	DDR address	RW	0xFFFF_FFFF
TRIG	0x0000_0000	Base+0x0064	SFC trigger	RW	0x0000_0007
SR	0x0000_0000	Base+0x0068	Status register	R	0x007F_007F
SCR	0x0000_0000	Base+0x006c	Status clear register	RW	0x0000_003F
INTC	0x0000_001F	Base+0x0070	Interrupt control	RW	0x0000_001F
FSM	0x0001_0849	Base+0x0074	Sub-modules' FSM state	R	0x000F_FBFF
CGE	0x0000_003F	Base+0x0078	CLK gate function	RW	0x0000_003F
DR	0x0000_0000	Base+0x1000	Data register (slave mode)	RW	0xFFFF_FFFF

10.7.3 Registers

10.7.3.1 SFC Global Register

	SFC_GLB																Base + 0x0000													
Bit	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
	Reserved																		TRAN_DIR	THRESHOLD						OP_MODE	PHASE_NUM			WP_EN	BURST_LEN	
RST	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	0	0	0	1	1	1	1

Bits	Name	Description	RW
31:14	Reserved	Write has no effect. Read as zero.	R
13	TRAN_DIR	The direction of transfer. 0: read data from SPI flash (default) 1: write data to SPI flash	RW
12:7	THRESHOLD	FIFO threshold. (unit: word) Natural number, 1-63(default is 1). Note: cannot be 0	RW
6	OP_MODE	0: slave mode (default) 1: DMA mode	RW
5:3	PHASE_NUM	The number of phase in one flow. 001: 1 phase (default) 010: 2 phases 011: 3 phases 100: 4 phases	RW

		101: 5 phases 110: 6 phases Note: cannot be 0	
2	WP_EN	Enable the hardware write protection 0: disable WP 1: Enable WP (default)	RW
1:0	BURST_MD	The max beats of burst. 00: INCR4 01: INCR8 10: INCR16 11: INCR32 (default)	RW

10.7.3.2 SFC Device Configure Register

	SFC_DEV_CONF																Base + 0x0004													
Bit	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
	Reserved														SMP_DELAY		CMD_TYPE	STA_TYPE		THOLD		TSETUP		TSH				CPHA	CPOL	CE_DL	HOLD_DL	WP_DL
RST	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	0	0	1	1	1

Bits	Name	Description	RW
31:18	Reserved	Write has no effect. Read as zero.	R
17:16	SMP_DELAY	Sample delay, it's depends on the device output delay, PCB delay and the IO ports delay, etc. 00: no delay (default) 01: half SFC_CLK cycle delay 10: one SFC_CLK cycle delay 11: one and a half SFC_CLK cycle delay	RW
15	CMD_TYPE	0: 8bits command type (default) 1: 16bits command type	RW
14:13	STA_TYPE	used for polling 00: 1 byte status (default) 01: 2 bytes status 10: 3 bytes status 11: 4 bytes status	RW
12:11	THOLD	Hold time of SFC_CE, the time between the last edge of SFC_CLK and SFC_CE de-assertion. 00: half SFC_CLK cycle (default) 01: one SFC_CLK cycle 10: one and a half SFC_CLK cycle	RW

		11: two SFC_CLK cycle	
10:9	TSETUP	Setup time of SFC_CE, the time between SFC_CE assertion and the first edge of SFC_CLK. 00: half SFC_CLK cycle (default) 01: one SFC_CLK cycle 10: one and a half SFC_CLK cycle 11: two SFC_CLK cycle	RW
8:5	TSH	The min interval time between twice assertion of SFC_CE. 0000: one SFC_CLK cycle 0001: one SFC_CLK cycle(default) 0010: one and a half SFC_CLK cycle 0011: two SFC_CLK cycle 0100: two and a half SFC_CLK cycle 0101: three SFC_CLK cycle 0110: three and a half SFC_CLK cycle 0111: four SFC_CLK cycle 1000: four and a half SFC_CLK cycle 1001: five SFC_CLK cycle 1010: five and a half SFC_CLK cycle 1011: six SFC_CLK cycle 1100: six and a half SFC_CLK cycle 1101: seven SFC_CLK cycle 1110: seven and a half SFC_CLK cycle 1111: eight SFC_CLK cycle	RW
4	CPHA	This bit is used to select the SPI clock format. 0 = Sampling of data occurs at odd edges (1,3,5,...) of the SFC_CLK clock (default) 1 = Sampling of data occurs at even edges (2,4,6,...) of the SFC_CLK clock	RW
3	CPOL	This bit selects the active level of SFC_CLK. 0 = Active-high clocks selected. In idle state SFC_CLK is low. (default) 1 = Active-low clocks selected. In idle state SFC_CLK is high.	RW
2	CE_DL	The default invalid level of SFC_CE pin 0: low 1: high (default)	RW
1	HOLD_DL	The default invalid level of SFC_HOLD pin 0: low 1: high (default)	RW
0	WP_DL	The default invalid level of SFC_WP pin 0: low 1: high (default)	RW

10.7.3.3 SFC Device Status Expected Register

	SFC_DEV_STA_EXP																Base + 0x0008															
Bit	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
	DEV_STA_EXP																															
RST	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

Bits	Name	Description	RW
31:0	DEV_STA_EXP	The expected value of flash status for comparing.	RW

10.7.3.4 SFC Device Status Real-time Register

	SFC_DEV_STA_RT																Base + 0x000c															
Bit	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
	DEV_STA_RT																															
RST	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

Bits	Name	Description	RW
31:0	DEV_STA_RT	The real-time status of SPI Flash	R

10.7.3.5 SFC Device Status Mask Register

	SFC_DEV_STA_MSK																Base + 0x0010															
Bit	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
	DEV_STA_MSK																															
RST	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

Bits	Name	Description	RW
31:0	DEV_STA_MSK	Flash status mask for comparing. 1: this bit will be checked. 0: this bit will not be checked.	RW

10.7.3.6 SFC Transfer Configure Register (phase0-phase5)

	SFC_TRAN_CONF0-5																Base + 0x0014 - Base + 0x0028															
Bit	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
	TRAN_MODE			ADDR_WIDTH			POLL_EN	CMD_EN	PHASE_FORMAT	DMY_BITS						DATA_EN	TRAN_CMD																
RST	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

Bits	Name	Description	RW
31:29	TRAN_MODE	Transfer mode: 000：Standard SPI（default） 001：Dual Input/Dual Output SPI 010：Dual I/O SPI 011：Full Dual I/O SPI 100：reserved 101：Quad Input/Quad Output SPI 110：Quad I/O SPI 111：Full Quad I/O SPI	RW
28:26	ADDR_WIDTH	The width of address used in the phase. 000: no address transfer (default) 001: 1 bytes 010: 2 bytes 011: 3 bytes 100: 4 bytes 101: 5 bytes >=110: 6 bytes	RW
25	POLL_EN	The polling function means that SFC checking the flash device's status. 0: no polling transfer (default) 1: has polling transfer. Note: it's priority is higher than tran_dir and data_en	RW
24	CMD_EN	0: no command transfer(default) 1: has command transfer	RW
23	PHASE_FORMAT	Phase format, see chapter 10.8.1 0: format0(default) 1: format1	RW
22:17	DMY_BITS	The needed dummy cycle num (unit: SFC_CLK cycle) 000000: no dummy transfer (default) ..... 000100: 4 dummy cycles	RW

		…… 000110: 6 dummy cycles …… 001000: 8 dummy cycles …… 001010: 10 dummy cycles …… 010000: 16 dummy cycles …… 011000: 24 dummy cycles …… 100000: 32 dummy cycles	
16	DATE_EN	0: no data transfer(default) 1: has data transfer	RW
15:0	CMD	The transfer command	RW

10.7.3.7 SFC Transfer Length Register

	SFC_TRAN_LEN																Base + 0x002c															
Bit	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
	TRAN_LEN																															
RST	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

Bits	Name	Description	RW
31:0	TRAN_LEN	Number of bytes to be transferred. 32'h0: no data transfer Note: this register should be 0 when there isn't data phase in the flow, else the DMA or RMC will start.	RW

10.7.3.8 SFC Device Address Register (phase0 - phase5)

	SFC_DEV_ADDR0-5																Base + 0x0030 - Base + 0x0044															
Bit	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
	DEV_ADDR																															
RST	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

Bits	Name	Description	RW
31:0	DEV_ADDR	The low bytes of flash address for transfer.	RW

10.7.3.9 SFC Device Address Plus Register (phase0 - phase5)

	SFC_DEV_ADDR_PLUS0-5																Base + 0x0048 - Base + 0x005c															
Bit	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
	DEV_ADDR_PLUS																															
RST	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

Bits	Name	Description	RW
31:0	DEV_ADDR_PLUS	The high bytes of flash address for transfer.	RW

10.7.3.10 SFC Memory Address Register

	SFC_MEM_ADDR																Base + 0x0060															
Bit	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
	MEM_ADDR																															
RST	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

Bits	Name	Description	RW
31:0	MEM_ADDR	Memory address for DMA. Note: this register must be 0 when slave mode	RW

10.7.3.11 SFC Trigger Register

	SFC_TRIG																Base + 0x0064															
Bit	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
	Reserved																													FLUSH	STOP	START
RST	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

Bits	Name	Description	RW
31:3	Reserved	Write has no effect. Read as zero.	R

2	FLUSH	Write 1 to reset the FIFO. Write 0 has no effect. Note: FIFO will be flushed after start.	W
1	STOP	Write 1 to stop all transmission. Write 0 has no effect.	W
0	START	Write 1 to start the transmission Write 0 has no effect.	W

10.7.3.12 SFC Status Register

	SFC_SR																Base + 0x0068															
Bit	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
	Reserved									FIFO_NUM							Reserved									BUSY		END	TRAN_REQ	RECE_REQ	OVER	UNDER
RST	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

Bits	Name	Description	RW
31:17	Reserved	Write has no effect. Read as zero.	R
22:16	FIFO_NUM	The quantity of data in FIFO (unit: word)	R
15:7	Reserved	Write has no effect. Read as zero.	R
6:5	BUSY	00: the SFC is IDLE 01: the SFC is in transaction 10: the SFC is in reception 11: the SFC is waiting for transfer	R
4	END	Indicate whether the transfer is end 0: the transfer isn't end or SFC is IDLE 1: the transfer is end This bit will generate INT_END signal.	R
3	TRAN_REQ	0: no request 1: vacancy num in FIFO is more than or equal to the threshold. This bit will generate the INT_TREQ interrupt.	R
2	RECE_REQ	0: no request 1: data num in FIFO is more than or equal to the threshold. This bit will generate the INT_RREQ interrupt.	R
1	OVER	0: no over-run 1: FIFO is over-run. This bit will generate the INT_OVER interrupt.	R
0	UNDER	0: no error. 1: FIFO is under-run This bit will generate INT_UNDR signal.	R

10.7.3.13 SFC Status Clear Register

	SFC_SCR																											Base + 0x006c				
Bit	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
	Reserved																											CLR_END	CLR_TREQ	CLR_RREQ	CLR_OVER	CLR_RUNDR
RST	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

Bits	Name	Description	RW
31:5	Reserved	Write has no effect. Read as zero.	R
4	CLR_END	0: has no effect. 1: clear the SFC_SR.END flag and the INT_END.	W
3	CLR_TREQ	0: has no effect. 1: clear the SFC_SR.TREQ flag and the INT_TREQ	W
2	CLR_RREQ	0: has no effect. 1: clear the SFC_SR.RREQ flag and the INT_RREQ	W
1	CLR_OVER	0: has no effect. 1: clear the SFC_SR.OVER flag and the INT_OVER	W
0	CLR_UNDR	0: has no effect. 1: clear the SFC_SR.UNDER flag and the INT_UNDR	W

10.7.3.14 SFC Interrupt Controller Register

	SFC_INTC																											Base + 0x0070				
Bit	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
	Reserved																											MASK_END	MASK_TREQ	MASK_RREQ	MASK_OVER	MASK_UNDR
RST	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	1	1	1	1

Bits	Name	Description	RW
31:5	Reserved	Write has no effect. Read as zero.	R
4	MASK_END	0: the INT_END is not masked 1: the INT_END will be masked (default)	RW
3	MASK_TREQ	0: the INT_TREQ is not masked 1: the INT_TREQ will be masked (default)	RW

2	MASK_RREQ	0: the INT_RREQ is not masked 1: the INT_RREQ will be masked (default)	RW
1	MASK_OVER	0: the INT_OVER is not masked 1: the INT_OVER will be masked (default)	RW
0	MASK_UNDR	0: the INT_UNDR is not masked 1: the INT_UNDR will be masked (default)	RW

Note: When the interrupt flag is cleared, the next interrupt will be generated immediately, so long as the requirement is contented.

10.7.3.15 SFC FSM Register

	SFC_FSM																Base + 0x0074													
Bit	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
	Reserved												FSM_AHB				FSM_SPI					Reserved	FSM_CLK				FSM_DMAC			FSM_RMC		
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1

Bits	Name	Description	RW
31:20	Reserved	Write has no effect. Read as indeterminate value.	R
19:16	FSM_AHB	The AHB_master module's FSM state	R
15:11	FSM_SPI	The SPI_ctrl module's FSM state	R
10	Reserved	Write has no effect. Read as indeterminate value.	R
9:6	FSM_CLK	The clk_gen module's FSM state	R
5:3	FSM_DMAC	The DMAC module's FSM state	R
2:0	FSM_RMC	The RCM module's FSM state	R

10.7.3.16 SFC Clk_gate Enable register

	SFC_CGE																										Base + 0x0078					
Bit	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
	Reserved																										CGE_SFC_EN	CGE_FIFO_EN	CGE_DMA_EN	CGE_RMC_EN	CGE_SPI_EN	CGE_REG_EN
RST	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	1	1	1	1	1

Bits	Name	Description	RW
31:6	Reserved	Write has no effect. Read as indeterminate value.	R
5	CGE_SFC_EN	0: disable the flow_ctrl modle's clk_gate function 1: enable the flow_ctrl module's clk_gate function (default)	RW
4	CGE_FIFO_EN	0: disable the fifo module's clk_gate function 1: enable the fifo module's clk_gate function (default)	RW
3	CGE_DMA_EN	0: disable the dmac and ahb_mst module's clk_gate function 1: enable the dmac and ahb_mst module's clk_gate function (default)	RW
2	CGE_RMC_EN	0: disable the rmc module's clk_gate function 1: enable the rmc module's clk_gate function (default)	RW
1	CGE_SPI_EN	0: disable the spi and clk_gen module's clk_gate function 1: enable the spi and clk_gen module's clk_gate function (default)	RW
0	CGE_REG_EN	0: disable the reg module's clk_gate function 1: enable the reg module's clk_gate function (default)	RW

Note：please disable all the clock gate function（clear all bits of SFC_CGE）

10.7.3.17 SFC Date Register

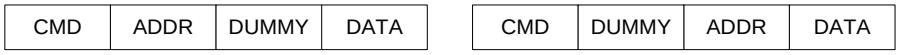
	SFC_DR																Base + 0x1000															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DR																															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:0	DR	Data register for FIFO.	RW

10.8 Software Guideline

10.8.1 Phase Description

- ✧ The phase is consisted of four sections: CMD, ADDR, DUMMY, DATA.
- ✧ Each phase could have one or more sections, four at the most.
- ✧ For one section, one phase could have one at the most.
- ✧ In all phases, only one phase can have DATA section.(we call this phase as data phase, other are command phase)
- ✧ Sections in a phase will be operated by two sequence format:



Format0 Format1

✧ For one flow, only one kind of the direction of transmission is supported.

10.8.2 Multi phases flow

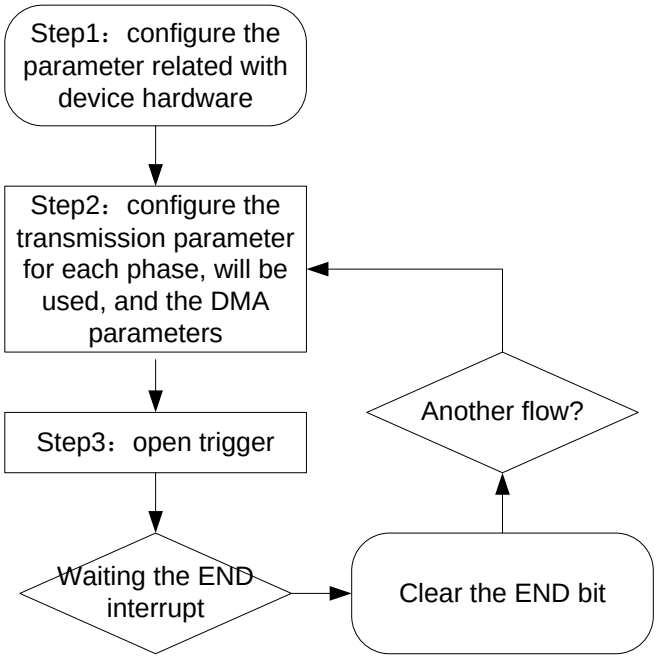


Figure 10-8 Multi phases flow

1 Configure the parameter related to the device hardware.
Such as the default level of IO port, timing parameter, etc.
related reg：SFC_DEV_CONF

2 Configure the parameter related to this transmission flow.
Such as num of phase, command format, operation mode, source address and target address, transfer direction and length, etc.
related reg：SFC_GLB、SFC_DEV_STA_EXP、SFC_DEV_STA_MSK、SFC_TRAN_CONF0-6、SFC_TRAN_LEN、SFC_DEV_ADDR0-6、SFC_MEM_ADDR

3 Open the trigger
related reg：SFC_TRIG

4 Waiting for end_end

5 Repeat step 2 for other flow

10.8.3 One phase flow

The flow can also be executed by only one phase. The NAND write perspective flow shown in the

Figure 10-9 as follow:

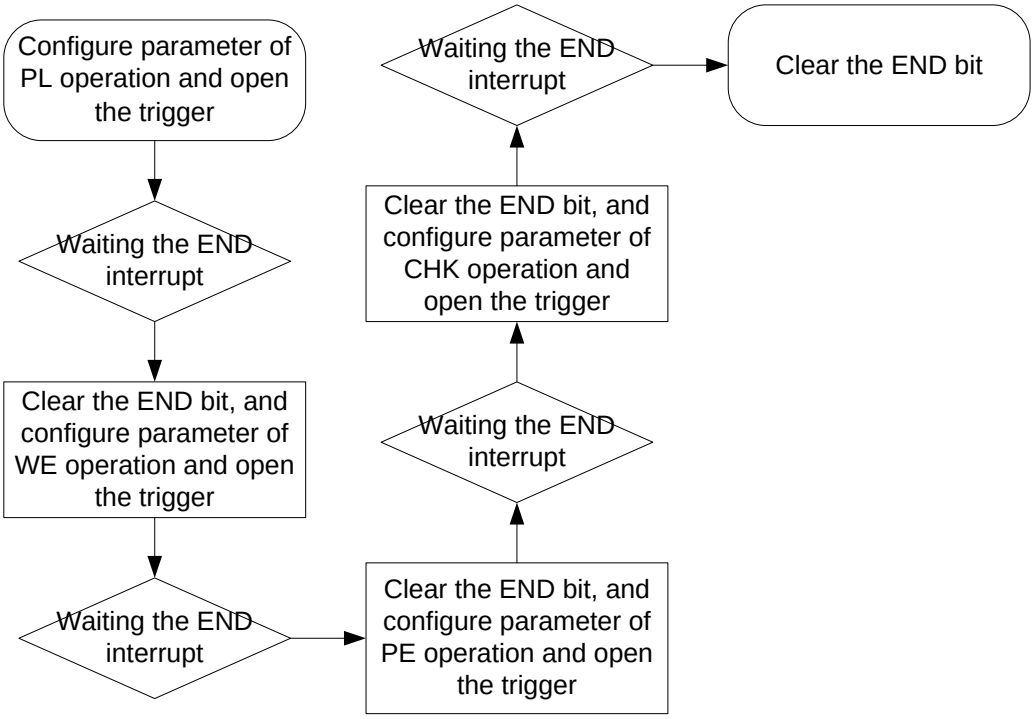


Figure 10-9 One phase flow

10.8.4 Meters of DMA operation need attention

DMA need some other parameter on the basic configuration.

- configure SFC_GLB (BURST_LEN、OP_MODE、BURST_LEN)
- configure SFC_MEM_ADDR (the memory address)

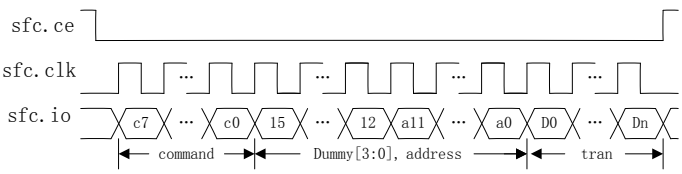
10.8.5 Meters of slave mode operation need attention

- For each data transfer phase, there must a flush operation before phase starting.
- The SFC will generate request according to the THRESHOLD. When CPU polling the SFC_SR register or waiting an interrupt, the number of data, that transferred by AHB bus each time must match the THRESHOLD.

10.8.6 Example (NAND flash write with multi phases in DMA mode)

a) configure SFC_DEV_CONF and SFC_GLB register

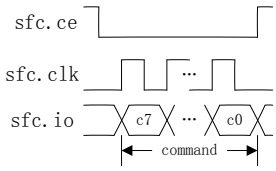
b) program load (PL)



Configure the parameter of PL into phase0 registers:

- configure the SFC_TRAN_CONF0(TRAN_MODE, CMD_EN, ADDR_WIDTH, DMY_NUM, etc)
- configure SFC_DEV_ADDR0
- configure SFC_TRAN_LEN

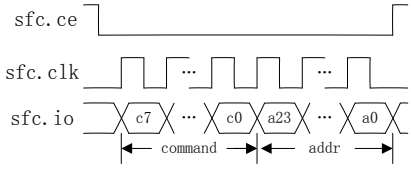
c) write enable (WE)



Configure the parameter of WE into phase1 registers:

- configure SFC_TRAN_CONF1(TRAN_MODE, CMD_EN, CMD, etc)

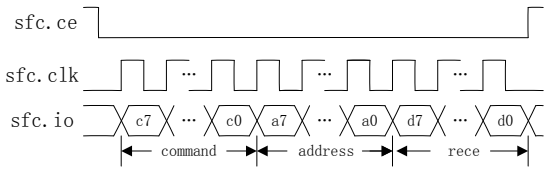
d) program execute



Configure the parameter of PE into phase2 registers:

- configure SFC_TRAN_CONF2(TRAN_MODE, CMD_EN, ADDR_WIDTH, CMD, etc)
- configure SFC_DEV_ADDR2

e) get features command to read the status



Configure the parameter of PE into phase3 registers:

- configure SFC_TRAN_CONF3(TRAN_MODE, CMD_EN, ADDR_WIDTH, CMD, POLL_EN, etc)
- configure SFC_DEV_ADDR3
- configure SFC_DEV_STA_EXP
- configure SFC_DEV_STA_MSK

f) Start

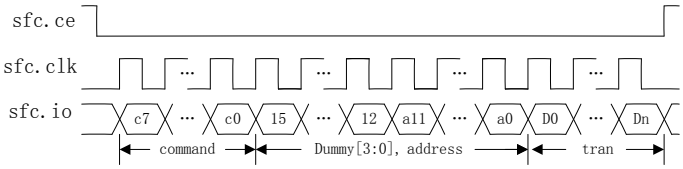
g) waiting the END interrupt

h) clear the END bit zone

10.8.7 Example (NAND flash write with single phase in REG mode)

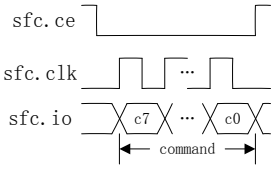
a) configure SFC_DEV_CONF and SFC_GLB register

b) program load (PL)



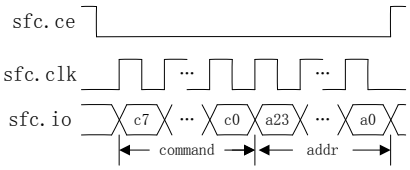
i. Set SFC_CTRL.FLUSH
ii. Configure the parameter of PL into phase0 registers:
– configure the SFC_TRAN_CONF0(TRAN_MODE, CMD_EN, ADDR_WIDTH, DMY_NUM, etc)
– configure SFC_DEV_ADDR0
– configure SFC_TRAN_LEN
– configure SFC_GLB.TRANS_DIR
iii. Start
iv. waiting SFC_SR.TREQ and clear, then set data to FIFO according to the THRESHOLD. Looping until all data trans end.
v. Waiting END and clear.

c) write enable (WE)



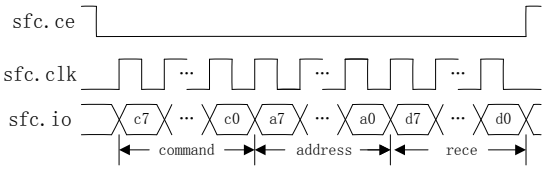
i. Configure the parameter of WE into phase0 registers:
– configure SFC_TRAN_CONF0(TRAN_MODE, CMD_EN, CMD, etc)
ii. Start
iii. Waiting END and clear.

d) program execute



i. Configure the parameter of PE into phase0 registers:
– configure SFC_TRAN_CONF0(TRAN_MODE, CMD_EN, ADDR_WIDTH, CMD, etc)
– configure SFC_DEV_ADDR0
ii. Start
iii. Waiting END and clear.

e) get features command to read the status



i. Configure the parameter of PE into phase0 registers:
   - configure SFC_TRAN_CONF0(TRAN_MODE, CMD_EN, ADDR_WIDTH, CMD, etc)
   - configure SFC_DEV_ADDR0
   - configure SFC_TRANS_LEN
   - configure SFC_GLB.TRANS_DIR

ii. Start

iii. waiting SFC_SR.RREQ and clear, then get data from FIFO according to the THRESHOLD. Looping until all data trans end.

iv. Waiting END and clear.

10.9 Index

Table 10-3 Terms and Abbreviations

Name	Description
AHB	Advanced Microcontroller Bus Architecture
Dual SPI	See SPI
CPU	Central Processing Unit
DMA	Direct Memory Access
Endian	The convention used to interpret the bytes making up a data word when those bytes are stored in computer memory. See Endianness at Wikipedia
FIFO	First In, First Out, a method for organizing and manipulating a data buffer
LSB	See Endian
MSB	See Endian
Quad SPI	See SPI
SPI	Serial Peripheral Interface
Standard SPI	See SPI
tCHSH	Time from clock pull high to select pull low. See a NOR Flash device spec
tSHSL	Time from select pull high to select pull low. See a NOR Flash device spec
tSLCH	Time from select pull low to clock pull high. See a NOR Flash device spec

Section 6

SYSTEM FUNCTIONS

11 Clock Reset and Power Controller

11.1 Overview

The Clock & Power management block consists of three parts: Clock control, PLL control, and Power control, Reset control.

The Clock control logic can generate the required clock signals including CCLK for CPU, HHCLK for L2CACHE, H0CLK for the AHB0, H2CLK for the AHB2, PCLK for the APB bus peripherals , DDR_CLK for DDR.

The Chip has four Phase Locked Loops (PLL):

1. APLL mainly is used to CPU and L2CACHE, also may use the AHB0 and AHB2, or other peripherals
2. MPLL mainly is used to DDR, also uses AHB0 and AHB2 APB, or other peripherals

For the power control logic, there are various power management schemes to keep optimal power consumption for a given task. The power management block can activate four modes: NORMAL mode, IDLE mode, SLEEP mode.

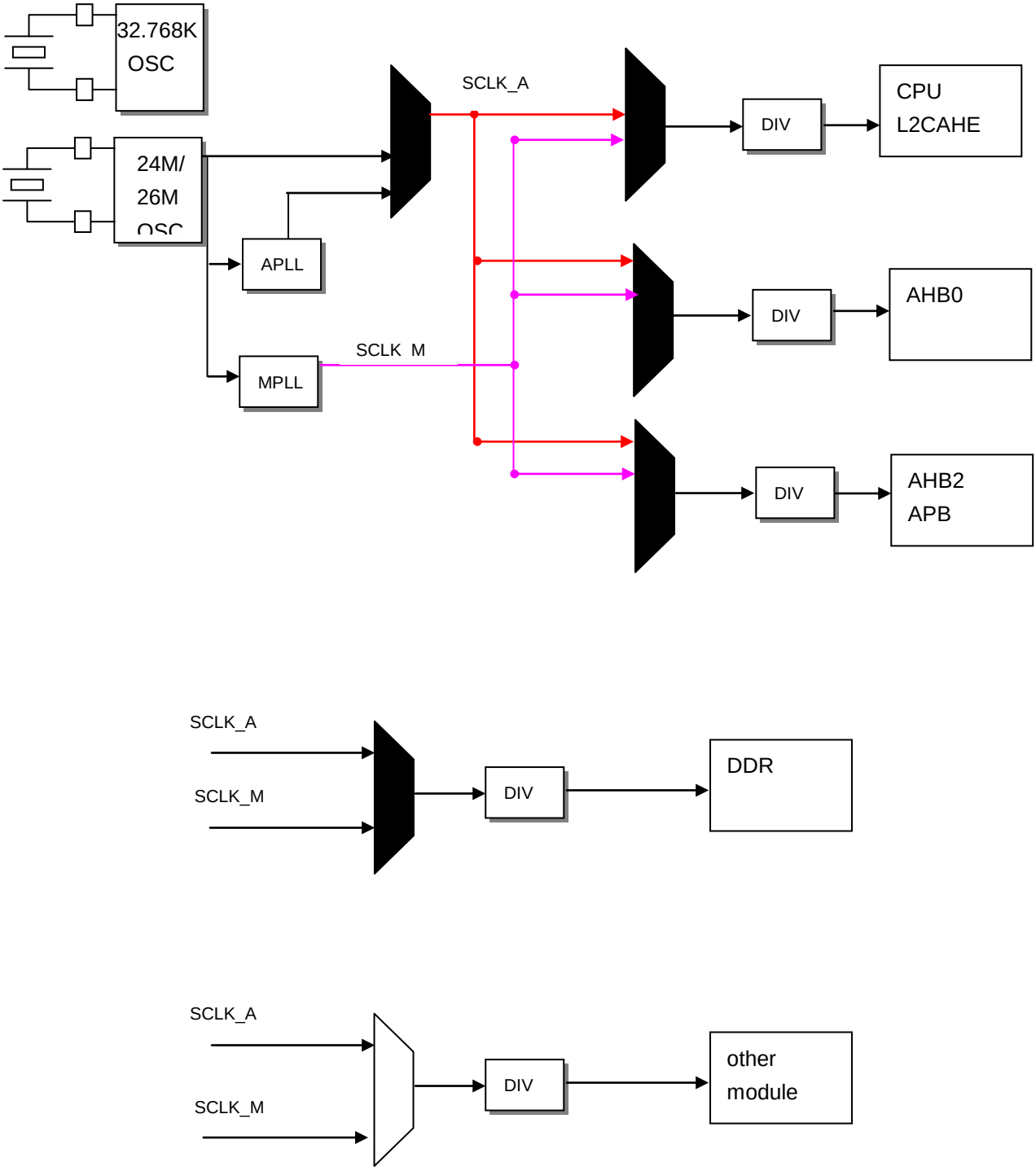
For reset control logic, the hardware reset and hibernate reset is extended to more 80ms. It controls or distributes all of the system reset signals.

Feature:

- On-chip 24/26MHz oscillator circuit
- On-chip 32.768KHZ oscillator circuit
- One two-chip phase-locked loops (PLL) with programmable multiplier
- CCLK, HHCLK, H2CLK, PCLK, H0CLK, DDR_CLK frequency can be changed separately for software by setting registers
- MSC clock supports 50M clock
- Functional-unit clock gating

11.1.1 CGU Block Diagram

Following figure illustrates a block diagram of CGU.



Note:

Clock mux **in grey color** represents **glitch-free** clock mux, which is free of glitches if clock selection is changed. Clock mux in white color represents non-glitch-free clock mux, which can suffer from glitches when changing clock sources. **Care must be taken in using each of clock muxes**.

11.1.2 CGU Registers

Following chapter will descript the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute
   R - Read only
   W - Write only
   RW - read and write
   RCW - read and write, but clear to 0 by read
   RSW - read and write, but set to 1 by read
   RWC - read and write, clear to 0 by write 1, write 0 has no effect
   RWS - read and write, set to 1 by write 1, write 0 has no effect
   RC - read only, and clear to 0 by read
   RS - read only, and set to 1 by read
   SPEC - special access method, relate to its description
3. Reset Value
   1 - reset to 1
   0 - reset to 0
   ? - value unknown after reset

Table 11-1 Registers Memory Map-Address Base

Name	Base	Description
CPM	0x10000000	Address base of CPM

Table 11-2 CGU Registers Configuration

Name	description	RW	Reset Value	Address offset	Access Size
CPCCR	Clock Control Register	RW	0x95000000	0x0000	32
CPCSR	Clock status register	RW	0x00000000	0x00D4	32
DDCDR	DDR clock divider Register	RW	0x40000000	0x002C	32
MACPHYCDR	MACPHY divider Register	RW	0x00000000	0x0054	32
I2SCDR	I2S device clock divider Register	RW	0x00000000	0x0060	32
I2SCDR1	I2S device clock divider Register	RW	0x00000000	0x0070	32
LPCDR	LCD pix clock divider Register	RW	0x00000000	0x0064	32
MSC0CDR	MSC0 clock divider Register	RW	0x40000000	0x0068	32
MSC1CDR	MSC1 clock divider Register	RW	0x00000000	0x00A4	32

USBCDR	USB clock divider Register	RW	0x00000000	0x0050	32
SSICDR	SSI clock divider Register	RW	0x00000000	0x0074	32
CIMCDR	CIM MCLK clock divider Register	RW	0x00000000	0x007C	32
PCMCDR	PCM device clock divider Register	RW	0x00000000	0x0084	32
PCMCDR1	PCM device clock divider Register	RW	0x00000000	0x00E0	32
MACPHYC	MAC PHY Control Register	RW	0x00000000	0x00E8	32
CPM_INTR	CPM interrupt Register	RW	0x00000000	0x00B0	32
CPM_INTRE	CPM interrupt Enable Register	RW	0x00000000	0x00B4	32
DRCG	DDR clk gate register	RW	0x00000001	0x00D0	32
CPSPR	CPM Scratch Pad Register	RW	0x????????	0x0034	32
CPSPPR	CPM Scratch Protected Register	RW	0x0000a5a5	0x0038	32
USBPCR	USB Parameter control register	RW	0x409919b8	0x003C	32
USBRDT	USB Reset Detect Timer Register	RW	0x02000096	0x0040	32
USBVBFIL	USB jitter filter Register	RW	0x00ff0080	0x0044	32
USBPCR1	USB Parameter control register 1	RW	0x00000004	0x0048	32
CPAPCR	APLL Control Register	RW	0x00000000	0x0010	32
CPMPCR	MPLL Control Register	RW	0x00000000	0x0014	32

11.1.2.1 Clock Control Register

The Clock Control Register (CPCCR) is a 32-bit read/write register, which controls CCLK, H0CLK, H2CLK and PCLK division ratios. It is initialized to 0x95000000 by any reset. Only word access can be used on CPCCR.

	CPCCR																													BASE+0x0000	
Bit	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
	SEL_SRC		SEL_CPLL		SEL_H0PLL		SEL_H2PLL		GATE_SCLKA	CE_CPU	CE_AHB0	CE_AHB2	PDIV				H2DIV				H0DIV				L2CDIV				CDIV			
RST	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:30	SEL_SRC	00: stop MUX clock output 01: EXCLK 10: APLL The mux clock output is SCLK_A The clock mux is free of glitches if clock selection is changed When switching clock source, it should be ensured that the clock switched from and the clock switched to both are running.	RW
29:28	SEL_CPLL	00: stop MUX clock output	RW

<table>
<tr><td></td><td></td><td>01: SCLK_A
10: MPLL
The mux clock output is to CPU and L2CAHE
The clock mux is free of glitches if clock selection is changed
When switching clock source, it should be ensured that the clock switched from and the clock switched to both are running.</td><td></td></tr>
<tr><td>27:26</td><td>SEL_H0PLL</td><td>00: stop MUX clock output
01: SCLK_A
10: MPLL
The mux clock output is to AHB0
The clock mux is free of glitches if clock selection is changed
When switching clock source, it should be ensured that the clock switched from and the clock switched to both are running.</td><td>RW</td></tr>
<tr><td>25:24</td><td>SEL_H2PLL</td><td>00 : stop MUX clock output
01: SCLK_A
10: MPLL
The mux clock output is to AHB2
The clock mux is free of glitches if clock selection is changed
When switching clock source, it should be ensured that the clock switched from and the clock switched to both are running.</td><td>RW</td></tr>
<tr><td>23</td><td>GATE_SCLKA</td><td>0: SCLK_A not gating
1: SCLK_A is gating.
Gating modules: LCD MSC MAC AIC USB SSI PCM CIM etc</td><td>RW</td></tr>
<tr><td>22</td><td>CE_CPU</td><td>Change enable for CPU and L2CAHE. If CE_CPU is 1 , write on CDIV,
L2CDIV will start a frequency changing sequence immediately. If CE_CPU is 0, writes on CDIV and L2CDIV have no affect.</td><td>RW</td></tr>
<tr><td>21</td><td>CE_AHB0</td><td>Change enable for AHB0. If CE_AHB0 is 1 , write on H0DIV will start a frequency changing sequence immediately. If CE_AHB0 is 0, writes on H0DIV have no affect.</td><td>RW</td></tr>
<tr><td>20</td><td>CE_AHB2</td><td>Change enable for AHB2. If CE_AHB2 is 1 , write on H2DIV, PDIV will start a frequency changing sequence immediately. If CE_AHB2 is 0, writes on H2DIV, PDIV have no affect.</td><td>RW</td></tr>
<tr><td>19:16</td><td>PDIV</td><td>Divider for Peripheral Frequency. Specified the PCLK division ratio.
<table>
<tr><th colspan="4">Bit 19~16: PDIV</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>X1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>X1/2</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>X1/3</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>X1/4</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>X1/5</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>X1/6</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>X1/7</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>X1/8</td></tr>
</table></td><td>RW</td></tr>
</table>

			1	0	0	0	X1/9
			1	0	0	1	X1/10
			1	0	1	0	X1/11
			1	0	1	1	X1/12
			1	1	0	0	X1/13
			1	1	0	1	X1/14
			1	**1**	**1**	**0**	**X1/X15**
			1	**1**	**1**	**1**	**X1/16**
15:12	H2DIV	Divider for AHB2 Frequency. Specified the AHB2 CLK division ratio.					RW
			Bit 15~12: H2DIV				Description
			0	0	0	0	X1
			0	0	0	1	X1/2
			0	0	1	0	X1/3
			0	0	1	1	X1/4
			0	1	0	0	X1/5
			0	1	0	1	X1/6
			0	1	1	0	X1/7
			0	1	1	1	X1/8
			1	0	0	0	X1/9
			1	0	0	1	X1/10
			1	0	1	0	X1/11
			1	0	1	1	X1/12
			1	1	0	0	X1/13
			1	1	0	1	X1/14
			1	**1**	**1**	**0**	**X1/15**
			1	**1**	**1**	**1**	**X1/16**
11:8	H0DIV	Divider for AHB0 Frequency. Specified the AHB0 CLK division ratio.					RW
			Bit 11~8: H0DIV				Description
			0	0	0	0	X1
			0	0	0	1	X1/2
			0	0	1	0	X1/3
			0	0	1	1	X1/4
			0	1	0	0	X1/5
			0	1	0	1	X1/6
			0	1	1	0	X1/7
			0	1	1	1	X1/8
			1	0	0	0	X1/9
			1	0	0	1	X1/10
			1	0	1	0	X1/11
			1	0	1	1	X1/12
			1	1	0	0	X1/13
			1	1	0	1	X1/14

			1	**1**	**1**	**0**	**X1/15**	
			1	**1**	**1**	**1**	**X1/16**	
7:4	L2CDIV	Divider for L2CACHE Frequency. Specified the L2C CLK division ratio.						RW
			Bit 7~4: L2CDIV				Description	
			0	0	0	0	X1	
			0	0	0	1	X1/2	
			0	0	1	0	X1/3	
			0	0	1	1	X1/4	
			0	1	0	0	X1/5	
			0	1	0	1	X1/6	
			0	1	1	0	X1/7	
			0	1	1	1	X1/8	
			1	0	0	0	X1/9	
			1	0	0	1	X1/10	
			1	0	1	0	X1/11	
			1	0	1	1	X1/12	
			1	1	0	0	X1/13	
			1	1	0	1	X1/14	
			1	**1**	**1**	**0**	**X1/15**	
			1	**1**	**1**	**1**	**X1/16**	
3:0	CDIV	Divider for CPU Frequency. Specified the CPU CLK division ratio.						
			Bit 3~0: CDIV				Description	
			0	0	0	0	X1	
			0	0	0	1	X1/2	
			0	0	1	0	X1/3	
			0	0	1	1	X1/4	
			0	1	0	0	X1/5	
			0	1	0	1	X1/6	
			0	1	1	0	X1/7	
			0	1	1	1	X1/8	
			1	0	0	0	X1/9	
			1	0	0	1	X1/10	
			1	0	1	0	X1/11	
			1	0	1	1	X1/12	
			1	1	0	0	X1/13	
			1	1	0	1	X1/14	
			1	**1**	**1**	**0**	**X1/15**	
			1	**1**	**1**	**1**	**X1/16**	

Import Note1 : for PDIV and AHB2DIV, AHB2's clock frequency must be 1 or 2 times of PDIV's clock frequency. For L2CDIV and CDIV, CPU's clock frequency must be 1,2,3, or 4 times of

L2CACHE's clock frequency.

Import Note2 : After reset, H0DIV and CE_AHB0 will be zero, and ahb0's clock freq and apb0's clock freq will be 1:1, but if you have ever set CE_AHB0 and change the value of H0DIV, ahb0's clock freq and apb0's clock freq will be 2:1 from then on.

11.1.2.2 Clock Status Register

Clock status register (CPCSR) is a 32-bit read/write register that specifies the status of CPCCR .This register is initialized to 0x00000000 only by any reset. Only word access can be used on CPCSR.

	CPCSR																													BASE+0x00D4		
Bit	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
	SRC_MUX	CPU_MUX	AHB0_MUX	AHB2_MUX	DDR_MUX	Reserved																								H2DIV_BUSY	H0DIV_BUSY	CDIV_BUSY
RST	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

Bits	Name	Description	RW
31	SRC_MUX	0: it indicates that mux switch has not stable 1: it indicates that mux switch has stable Software must check the value, then go next	R
30	CPU_MUX	0: it indicates that mux switch has not stable 1: it indicates that mux switch has stable Software must check the value, then go next	R
29	AHB0_MUX	0: it indicates that mux switch has not stable 1: it indicates that mux switch has stable Software must check the value, then go next	R
28	AHB2_MUX	0: it indicates that mux switch has not stable 1: it indicates that mux switch has stable Software must check the value, then go next	R
27	DDR_MUX	0: it indicates that mux switch has not stable 1: it indicates that mux switch has stable Software must check the value, then go next	R
26:3	Reserved	Writing has no effect, read as zero.	R
2	H2DIV_BUSY	The bit is a read-only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going Software should be wait until H2DIV_BUSY == 0, then may begin another frequency change.	R
1	H0DIV_B	The bit is a read-only bit. It indicates whether the frequency change has	R

	USY	finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going Software should be wait until H0DIV_BUSY == 0, then may begin another frequency change.	
0	CDIV_BUSY	The bit is a read-only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going Software should be wait until CDIV_BUSY == 0, then may begin another frequency change.	R

11.1.2.3 DDR Memory clock divider Register

DDR memory clock divider Register (DDRCDR) is a 32-bit read/write register that specifies the divider of DDR memory clock . This register is initialized to 0x40000000 only by any reset. Only word access can be used on DDRCDR.

	DDRCDR																BASE+0x002C															
Bit	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
	DCS		CE_DDR	DDR_BUSY	DDR_STOP	GATE_EN	CHANGE_EN	FLAG	Reserved																				DDRCDR			
RST	0	1	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

Bits	Name	Description	RW
31:30	DCS	DDR Clock Source Selection. Selects the DDR clock source between APLL output and MPLL output. 00: stop MUX clock output 01: DDR clock source is SCLK_A 10: DDR clock source is MPLL The clock mux is free of glitches if clock selection is changed When switching clock source, it should be ensured that the clock switched from and the clock switched to both are running.	RW
29	CE_DDR	Change enable for DDR. If CE_DDR is 1 , write on DDRCDR will start a frequency changing sequence immediately. If CE_DDR is 0, writes on DDRCDR have no affect	RW
28	DDR_BUSY	The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going Software should be wait until DDR_BUSY == 0, then may begin another frequency change.	R
27	DDR_STOP	When DDR_STOP is 1 and CE_DDR is 1, the DDR clock will stop. When DDR_STOP is 0 and CE_DDR is 1, the DDR clock will continue.	RW

		Software should wait until DDR_BUSY == 0, then may begin another frequency change. DDR_STOP is prior to DDRCDR.	
26	GATE_EN	1'b1: if DDR gives a clock stop request to CPM, and CPM will gate DDR's clock.	
25	CHANGE_EN	When this bit set high, CPM will change DDR's clock when CE_DDR is high and FLAG is low	RW
24	FLAG	1：DDR reject frequency change 0: DDR not reject frequcy change	R
23:4	Reserved	Writing has no effect, read as zero.	R
3:0	DDRCDR	Divider for DDR Frequency. Specified the DDR memory clock division ratio, which varies from 1 to 16 (division ratio = DDRCDR + 1).	RW

11.1.2.4 MACPHY clock divider Register

MAC clock divider Register (MACCDR) is a 32-bit read/write register that specifies the divider of MAC PHY clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on MACCDR.

	MACCDR																														
																	BASE+0x0054														
Bit	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
	MACPCS	Reserved	CE_MAC	MAC_BUSY	MAC_STOP	Reserved											MACCDR															
RST	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

Bits	Name	Description	RW
31	MACPCS	MAC PHY Clock Source Selection. Selects the MAC PHY clock source 0: MAC PHY clock source is SCLK_A 1: MAC PHY clock source is MPLL It is no glitch free mux.	RW
30	Reserved	Writing has no effect, read as zero.	R
29	CE_MAC	Change enable for MAC. If CE_MAC is 1 , write on MACCDR ill start a frequency changing sequence immediately. If CE_MAC is 0, writes on MACCDR have no affect	RW
28	MAC_BUSY	The bit is read only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going Software should wait until MAC_BUSY == 0, may begin another frequency change.	R
27	MAC_STOP	When MAC_STOP is 1 and CE_MAC is 1, the MAC clock will stop. When MAC_STOP is 0 and CE_MAC is 1, the MAC clock will continue.	RW

		Software should wait until MAC_BUSY == 0, may begin another frequency change.	
26:8	Reserved	Writing has no effect, read as zero.	R
7:0	MACCDR	Divider for MAC PHY Frequency. Specified the MAC device clock division ratio division ratio = MACCDR + 1	RW

11.1.2.5 I2S device clock divider Register

I2S device clock divider Register (I2SCDR) is a 32-bit read/write register that specifies the divider of I2S device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on I2SCDR.

	I2SCDR																														BASE+0x0060
Bit	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
	I2PCS		CE_I2S	Reserved							I2SDIV_M						I2SDIV_N															
RST	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

Bits	Name	Description	RW
31	I2PCS	0: SCLK_A output 1: MPLL output	RW
30	I2CS	0: EXCLK 1: PLL output	
29	CE_I2S	AIC M/N divider(1) enable.	RW
28:22	Reserved	Writing has no effect, read as zero.	R
21:13	I2SDIV_M	Number M of AIC M/N divider(1).	RW
12:0	I2SDIV_N	Number N of AIC M/N divider(1).	RW

Note (1): AIC M/N divider is a decimal divider. The output clock frequency of AIC M/N divider is:

Fout = Fin * M / N.

N should not be less than M*2.

AIC M/N divider output clock duty cycle is possible not 50%. But adjusting the M and N value, which makes the duty cycle is close to 50%. Error of duty cycle is:

De_p = (ceill(N / M) - (N / M)) / (N / M). ceil: Round toward positive infinity.

and:

De_n = (floor(N / M) - (N / M)) / (N / M). floor: Round toward negative infinity.

Recommend the De_p is less than 5% and De_n is great than -5%.

If the parameter D of AIC M/N divider is in automatic calculation mode, namely I2S_DEN

is set LOW, it is recommended to refresh the value in I2SCDR1 after changing the parameter in I2SCDR. In another words, after changing the parameter in I2SCDR, read I2SCDR1 and set the same value to I2SCDR1.

11.1.2.6 I2S device clock divider Register 1

I2S device clock divider Register 1(I2SCDR1) is a 32-bit read/write register that specifies the divider of I2S device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on I2SCDR1.

	I2SCDR1																													BASE+0x0070	
Bit	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
	I2S_NEN	I2S_DEN	Reserved														I2SDIV_D															
RST	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

Bits	Name	Description	RW
31	I2S_NEN	0: The number N of AIC M/N divider is automatic calculate. 1: The number N of AIC M/N divider is set by I2SDIV_N. The automatic calculation is: $f_{out} = f_{in} * M/(N-0x1fff+M)$	RW
30	I2S_DEN	0: The number D of AIC M/N divider is automatic calculate. 1: The number D of AIC M/N divider is set by I2SDIV_D.	RW
29:13	Reserved	Writing has no effect, read as zero.	R
12:0	I2SDIV_D	The number D of AIC M/N divider(2).	RW

Note (2): General D = N/2. Normal we can keep I2S_DEN \ I2S_NEN to '0' and don't care 'I2SDIV_D'.

There is a little mistake in this register: the 30 and 31 bit is miss-connected. For example, write 30 bit to "1", but it will be reflected in 31 bit； write 31 bit to "1", but it will be reflected in 30 bit.

11.1.2.7 LCD pix clock divider Register

LCD pix clock divider Register (LPCDR) is a 32-bit read/write register that specifies the divider of LCD pixel clock (LPCLK). This register is initialized to 0x00000000 by any reset. Only word access can be used on LPCDR.

	LPCDR																													BASE+0x0064	
Bit	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
	LPCS	Reserved		CE_LCD	LCD_BUSY	LCD_STOP	Reserved																		LPCDR							
RST	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

Bits	Name	Description	RW
31	LPCS	0: SCLK_A output 1: MPLL output It is no glitch free mux. Software should be stop PIX clock, when change this bit.	RW
30	Reserved		
28	CE_LCD	Change enable for LCD. If CE_LCD is 1 , write on LPCDR will start a frequency changing sequence immediately. If CE_LCD is 0, writes on LPCDR have no affect	RW
27	LCD_BUSY	The bit is read only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going Software should be wait until LCD_BUSY == 0, may begin another frequency change.	R
26	LCD_STOP	When LCD_STOP is 1 and CE_LCD is 1, the LCD clock will stop. When LCD_STOP is 0 and CE_LCD is 1, the LCD clock will continue. Software should wait until LCD_BUSY == 0, may begin another frequency change. LCD_STOP is prior to LPCDR. If system don't use this module, may set this bit to reduce power .	RW
25:8	Reserved	Writing has no effect, read as zero.	RW
7:0	LPCDR	Divider for Pixel Frequency. Specified the LCD pixel clock (LPCLK) division ratio, which varies from 1 to 256 (division ratio = LPCDR + 1).	RW

11.1.2.8 MSC0 device clock divider Register

MSC0 device clock divider Register (MSC0CDR) is a 32-bit read/write register that specifies the divider of MSC0 device clock . This register is initialized to 0x40000000 only by any reset. Only word access can be used on MSC0CDR.

	MSC0CDR																													BASE+0x0068
Bit	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
	MPCS	Reserved	CE_MSC0	MSC0_BUSY	MSC0_STOP	Reserved											S_CLK0_SEL	Reserved								MSC0CDR							
RST	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

Bits	Name	Description	RW
31	MPCS	0: select SCLK_A clock output 1: select MPLL clock output	RW

		The clock mux is not free of glitches if clock selection is changed The clock mux output is to MSC0, MSC1 . It is source clock of MSC0, MSC1.	
30	Reserved		
29	CE_MSC0	Change enable for MSC0. If CE_MSC0 is 1 , write on MSC0CDR ill start a frequency changing sequence immediately. If CE_MSC0 is 0, writes on MSC0CDR have no affect	RW
28	MSC0_BUSY	The bit is read only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going Software should be wait until MSC0_BUSY == 0, may begin another frequency change.	R
27	MSC0_STOP	When MSC0_STOP is 1 and CE_MSC0 is 1, the MSC0 clock will stop. When MSC0_STOP is 0 and CE_MSC0 is 1, the MSC0 clock will continue. Software should wait until MSC0_BUSY == 0, may begin another frequency change.MSC0_STOP is prior to MSC0CDR. If system don't use this module, may set this bit to reduce power .	RW
26:16	Reserved	Writing has no effect, read as zero.	R
15	S_CLK0_SEL	MSC sample clock selection: 0: Sample clock is 90-degree phase shifted by device clock 1: Sample clock is 180-degree phase shifted by device clock Note:the driver clk is always 180-degree phase shifted by device clock	RW
14:8	Reserved		
7:0	MSC0CDR	Divider for MSC0 Frequency. Specified the MSC0 device clock division ratio division ratio = (MSC0CDR + 1)*2.	RW

11.1.2.9 MSC1 device clock divider Register

MSC1 device clock divider Register (MSC1CDR) is a 32-bit read/write register that specifies the divider of MSC1 device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on MSC1CDR.

	MSC1CDR																													BASE+0x00A4
Bit	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
	Reserved		CE_MSC1	MSC1_BUSY	MSC1_STOP	Reserved											S_CLK1_SEL	Reserved								MSC1CDR							
RST	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

Bits	Name	Description	RW
31:30	Reserved		R
29	CE_MSC1	Change enable for MSC1. If CE_MSC1 is 1 , write on MSC1CDR ill start a frequency changing sequence immediately. If CE_MSC1 is 0, writes on MSC1CDR have no affect	RW
28	MSC1_BUSY	The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going Software should be wait until MSC1_BUSY == 0, may go next	R
27	MSC1_STOP	When MSC1_STOP is 1 and CE_MSC1 is 1, the MSC1 clock will stop. When MSC1_STOP is 0 and CE_MSC1 is 1, the MSC1 clock will continue. Software should wait until MSC1_BUSY == 0, may begin another frequency change.MSC1_STOP is prior to MSC1CDR. If system don't use this module, may set this bit to reduce power .	RW
26:16	Reserved	Writing has no effect, read as zero.	R
15	S_CLK1_SEL	MSC1 sample clock selection: 0: Sample clock is 90-degree phase shifted by device clock 1: Sample clock is 180-degree phase shifted by device clock	RW
14:8	Reserved		
7:0	MSC1CDR	Divider for MSC1 Frequency. Specified the MSC1 device clock division ratio division ratio = (MSC1CDR + 1)*2.	RW

11.1.2.10 USB clock divider Register

USB clock divider Register (USBCDR) is a 32-bit read/write register that specifies the divider of USBPHY clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on USBCDR.

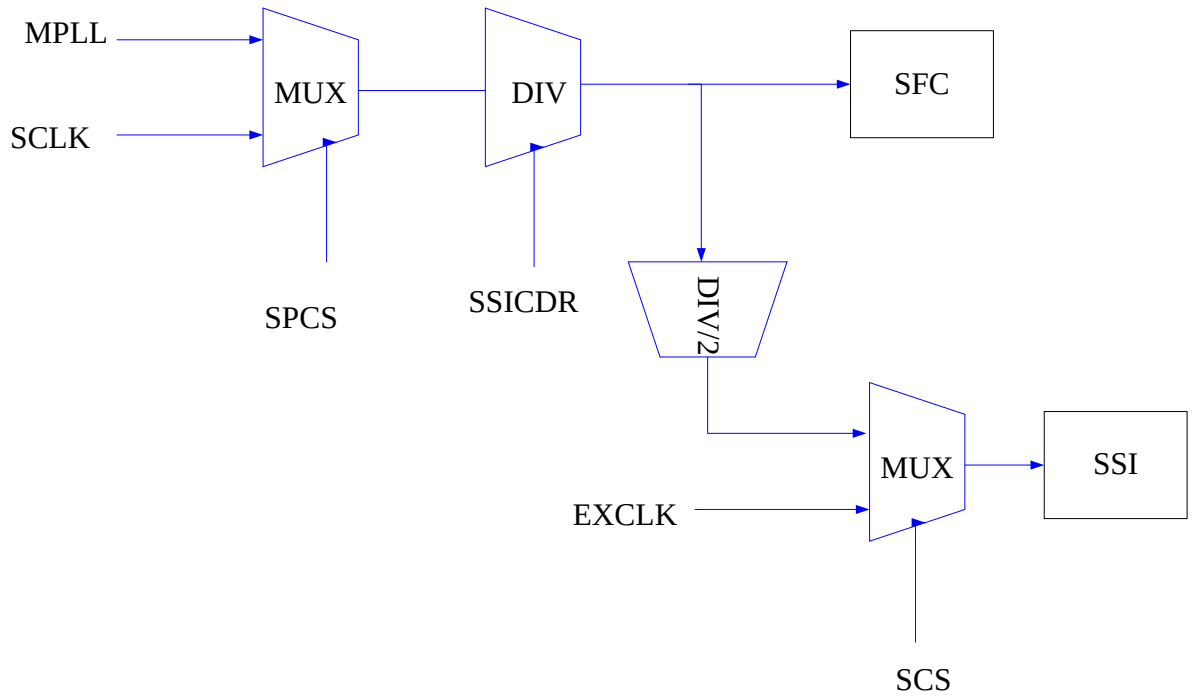
	USBCDR																														0x10000050
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UCS	UPCS	CE_USB	USB_BUSY	USB_STOP	Reserved																			USBCDR							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	UCS	USB Clock Source Selection. Selects the USB clock source between PLL output and pin EXCLK. 0: USB clock source is EXCLK 1: USB clock source is PLL output divided by USBCDR If UCS == 0, the clock divider is gated, don't set CE_USB	RW

		It is no glitch free mux. Software should be stop the module clock, when change this bit	
30	UPCS	0: USB clock source is SCLK_A output 1: USB clock source is MPLL output	
29	CE_UHC	Change enable for USB. If CE_USB is 1 , write on USBCDR ill start a frequency changing sequence immediately. If CE_USB is 0, writes on USBCDR have no affect	RW
28	USB_BUSY	The bit is read only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going Software should wait until USB_BUSY == 0, may begin another frequency change.	R
27	USB_STOP	When USB_STOP is 1 and CE_USB is 1, the USB clock will stop. When USB_STOP is 0 and CE_USB is 1, the USB clock will continue. Software should wait until USB_BUSY == 0, may begin another frequency change. USB_STOP is prior to USBCDR. If system don't use this module, may set this bit to reduce power .	RW
26:8	Reserved	Writing has no effect, read as zero.	R
7:0	USBCDR	Divider for USB Frequency. Specified the USB device clock division ratio division ratio = USBCDR + 1	RW

11.1.2.11 SFC device clock divider Register

SFC device clock divider Register (SSICDR) is a 32-bit read/write register that specifies the divider of SFC device clock . This register is initialized to 0x00000000 only by any reset. Only word access can be used on SSICDR.the clk relation ship of SFC and SSI is as follows:



	SSICDR																													BASE+0x0074
Bit	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
	SPCS	SCS	CE_SSI	SSI_BUSY	SSI_STOP	Reserved																			SSICDR							
RST	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

Bits	Name	Description	RW
31	SPCS	0: select SCLK_A output 1: select MPLL output It is direct to SFC module It is no glitch free mux. Software should be stop the module clock, when change this bit	RW
30	SCS	0: EXCLK 1: SFC/2 output It is direct to SSI module	
29	CE_SSI	Change enable for SSI. If CE_SSI is 1 , write on SSICDR will start a frequency changing sequence immediately. If CE_SSI is 0, writes on SSICDR have no affect	RW
28	SSI_BUSY	this bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going Software should be wait until SSI_BUSY == 0, may begin another	R

		frequency change.	
27	SSI_STOP	When SSI_STOP is 1 and CE_SSI is 1, the SSI clock will stop. When SSI_STOP is 0 and CE_SSI is 1, the SSI clock will continue. Software should wait until SSI_BUSY == 0, may begin another frequency change. SSI_STOP is prior to SSICDR. If system don't use this module, may set this bit to reduce power .	RW
26:8	Reserved	Writing has no effect, read as zero.	R
7:0	SSICDR	Divider for SSI Frequency. Specified the SSI device clock division ratio, which varies from 1 to 256 (division ratio = SSICDR + 1).	RW

11.1.2.12 CIM MCLK clock divider Register

CIM mclk clock divider Register (CIMCDR) is a 32-bit read/write register that specifies the divider of CIM mclk clock (CIM_MCLK). This register is initialized to 0x00000000 only by any reset. Only word access can be used on CIMCDR.

	CIMCDR																BASE+0x007C													
Bit	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
	CIMPCS	Reserved	CE_CIM	CIM_BUSY	CIM_STOP	Reserved																			CIMCDR							
RST	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	0	0

Bits	Name	Description	RW
31	CIMPCS	0: select SCLK_A clock output 1: select MPLL clock output It is no glitch free mux. Software should be stop the module clock, when change this bit	RW
30	Reserved		
29	CE_CIM	Change enable for CIM. If CE_CIM is 1 , write on CIMCDR ill start a frequency changing sequence immediately. If CE_CIM is 0, writes on CIMCDR have no affect	RW
28	CIM_BUSY	The bit is ready only bit. It indicates whether the frequency change has finished. When the bit is 0, it indicates frequency change has finished., otherwise it indicates the frequency change is on going Software should be wait until CIM_BUSY == 0, may begin another frequency change.	R

27	CIM_STOP	When CIM_STOP is 1 and CE_CIM is 1, the CIM clock will stop. When CIM_STOP is 0 and CE_CIM is 1, the CIM clock will continue. Software should wait until CIM_BUSY == 0, may begin another frequency change. CIM_STOP is prior to CIMCDR. If system don't use this module, may set this bit to reduce power .	RW
26:8	Reserved	Writing has no effect, read as zero.	R
7:0	CIMCDR	Divider for CIM MCLK Frequency. Specified the CIM MCLK clock (CIM_MCLK) division ratio, which varies from 1 to 256 (division ratio = CIMCDR + 1). the output clock is "io_setup.cim_mclk_o".	RW

11.1.2.13 PCM device clock divider Register

PCM device clock divider Register (PCMCDR) is a 32-bit read/write register that specifies the divider of PCM device clock . This register is initialized to 0x00002002 only by any reset. Only word access can be used on PCMCDR.

PCMCDR																														BASE+0x0084
Bit	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
	PCMCPCS		CE_PCM	Reserved							PCMDIV_M						PCMDIV_N															
RST	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

Bits	Name	Description	RW
31:30	PCMPCS	00: select SCLK_A output 01: EXCLK 10: select MPLL output 11: Reserved It is no glitch free mux. Software should be stop the module clock, when change this bit	RW
29	CE_PCM	PCM M/N divider(1) enable.	RW
28:22	Reserved	Writing has no effect, read as zero.	R
21:13	PCMDIV_M	Number M of AIC M/N divider(1).	RW
12:0	PCMDIV_N	Number N of AIC M/N divider(1).	RW

Note (1): AIC M/N divider is a decimal divider. The output clock frequency of AIC M/N divider is:

Fout = Fin * M / N.

AIC M/N divider output clock duty cycle is possible not 50%. But adjusting the M and N value, which makes the duty cycle is close to 50%. Error of duty cycle is:

De_p = (ceill(N / M) - (N / M)) / (N / M). ceil: Round toward positive infinity.

and:

De_n = (floor(N / M) - (N / M)) / (N / M). floor: Round toward negative infinity.
Recommend the De_p is less than 5% and De_n is great than -5%.

11.1.2.14 PCM device clock divider Register 1

PCM device clock divider Register 1 (PCMCDR1) is a 32-bit read/write register that specifies the divider of PCM device clock . This register is initialized to 0x00000001 only by any reset. Only word access can be used on PCMCDR.

	PCMCDR1																													BASE+0x00e0	
Bit	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
	PCM_NEN	PCM_DEN																		PCMDIV_D												
RST	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

Bits	Name	Description	RW
31	PCM_NEN	0: The number N of PCM M/N divider is automatic calculate. 1: The number N of PCM M/N divider is set by PCMDIV_N. The automatic calculation is: $f_{out} = f_{in} * M/(N-0x1fff+M)$	RW
30	PCM_DEN	0: The number D of PCM M/N divider is automatic calculate. 1: The number D of PCM M/N divider is set by PCMDIV_D.	RW
29:13	Reserved	Writing has no effect, read as zero.	R
12:0	PCMDIV_D	The number D of AIC M/N divider(2).	RW

Note (2): General D = N/2. Normal we can keep PCM_DEN \ PCM_NEN to '0' and don't care 'PCMDIV_D'.

There is a little mistake in this register: the 30 and 31 bit is miss-connected. For example, write 30 bit to "1", but it will be reflected in 31 bit； write 31 bit to "1", but it will be reflected in 30 bit.

11.1.2.15 MAC PHY Control Register (MPHYC)

The MAC PHY Control Register is a 32-bit read/write register that control MAC interface.It is initialized to 0x0000000 0by reset.

	MPHYC																													0x100000E8	
Bit	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
	Mode_sel	MAC_SPEED																											SOFT_RST	PHY_INTF		
RST	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

Bits	Name	Description	RW
31	Mode_sel	useless now	RW
30:29	MAC_SPEED	It is only read. 0x: 1000M 10: 10M 11: 100M	R
28:4	Reserved		R
3	SOFT_RST	1:reset MAC 0:don't reset MAC	RW
2:0	PHY_INTF	useless now	RW

11.1.2.16 CPM interrupt Register

	CPM_INTR																BASE+0x00B0															
Bit	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
	Reserved																														VBUS_INTR	ADEV_INTR
RST	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

Bits	Name	Description	RW
31:3	Reserved	Writing has no effect, read as zero.	R
1	VBUS_INTR	B device insert interrupt.	R
0	ADEV_INTR	A device insert interrupt.	R

11.1.2.17 CPM interrupt enable Register

	CPM_INTRE																BASE+0x00B4															
Bit	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
	Reserved																														VBUS_INTRE	ADEV_INTRE
RST	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

Bits	Name	Description	RW
31:3	Reserved	Writing has no effect, read as zero.	R

1	VBUS_INTRE	B device insert interrupt enable.	RW
0	ADEV_INTRE	A device insert interrupt enable.	RW

11.1.2.18 DDR Clk Gate Register

	DRCG																														BASE+0x00D0	
Bit	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
	Reserved																							DRCG								
RST	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

Note: Please refer to DDR and NFI's spec for detail use of this register.

11.1.2.19 CPM Scratch Pad Protected Register

The Scratch Pad Protected Register is reset to 0x0000a5a5. When CPSPPR value equals to 0x00005a5a, software can write the CPSPR. or else can't.

	CPSPPR																														BASE+0x0038	
Bit	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
	Reserved																CPSPPR															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1

Bits	Name	Description	RW
31:16	Reserved	Writing has no effect, read as zero.	R
15:0	CPSPPR	The value is only = 0x00005a5a, software can write the CPSPR.	RW

11.1.2.20 CPM Scratch Pad Register

The Scratch Pad Register is a 32-bit read/write register that allows software to preserve some critical data . It is not initialized by power on and WDT reset.

	CPPSR																														BASE+0x0034	
Bit	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
	CPPSR																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

11.1.2.21 USB Parameter Control Register

The USBPCR is a 32-bit read/write register that allows software to control OTG PHY some functions. It is initialized to 0x409919b8.

	USBPCR																0x1000003C													
Bit	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
	USB_MODE	AVLD_REG	IDPULLUP_MASK		INCR_MASK	TXRISETUNE	COMONONN	VBUSVLDEXT	VBUSVLDEXTSEL	POR	SIDDQ	OTGDISABLE	COMPDISTUNE				OTGTUNE			SQRXTUNE			TXFSLSTUNE				TXPREMTUNE	TXHSXVTUNE		TXVREFTUNE			
RST	0	1	0	0	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0	1	1	0	0	1	1	0	1	1	1	0	0	0

Bits	Name	Description	RW
31	USB_MODE	0: work as USB device 1: work as OTG	RW
30	AVLD_REG	This bit is used to set “avalid”(VBUS above A-device session threshold) signal.	RW
29:28	IDPULLUP_MASK	These 2 bits control “idpullup” signal in otg mode. 2’b1x: “idpullup” always active 2’b01: “idpullup” always active when usb suspend 2’b00: use “idpullup” from otg controller	RW
27	INCR_MASK	This bit controls whether the ahb interface enhancement for “incr transfer” takes effect. Set this bit to 0 will active the enhancement.	RW
26	TXRISETUNE	This bit adjusts the rise/fall times of the high-speed waveform 1: -8% 0: default	RW
25	COMMONONN	This bit is the OTG PHY common block power down control signal. 0: The common blocks remain powered in suspend mode 1: The common blocks are powered down in suspend mode	RW
24	VBUSVLDEXT	This bit controls OTG PHY VBUSVLDEXT signal.	RW
23	VBUSVLDEXTSEL	This bit controls OTG PHY VBUSVLDEXTSEL signal.	RW
22	POR	This bit controls OTG PHY power on reset.	RW
21	SIDDQ	This bit is the OTG PHY analog blocks power down signal.	RW
20	OTG_DISABLE	This bit is the power control for otg block in OTG PHY.	RW
19:17	COMPDISTUNE	These bits control disconnect threshold adjustment. 3’b111 +4.5% 3’b110 +3% 3’b101 +1.5% 3’b100 Default 3’b011 -1.5%	RW

		3'b010	-3%		
		3'b001	-4.5%		
		3'b000	-6%		
16:14	OTGTUNE	These bits control VBUS valid threshold adjustment.		RW	
		3'b111	+4.5%		
		3'b110	+3%		
		3'b101	+1.5%		
		3'b100	Default		
		3'b011	-1.5%		
		3'b010	-3%		
		3'b001	-4.5%		
		3'b000	-6%		
13:11	SQRXTUNE	These bits control squelch threshold adjustment.		RW	
		3'b111	-20%		
		3'b110	-15%		
		3'b101	-10%		
		3'b100	-5%		
		3'b011	default		
		3'b010	+5%		
		3'b001	+10%		
		3'b000	+15%		
10:7	TXFSLSTUNE	These bits control FS/LS source impedance adjustment.		RW	
		4'b1111	-5%		
		4'b0111	-2.5%		
		4'b0011	Default		
		4'b0001	+2.5%		
		4'b0000	+5%		
6	TXPREEMPHTUNE	This bit controls HS transmitter Pre-emphasis enable. 1: enable 0: disable		RW	
5:4	TXHSXVTUNE	These bits adjust the voltage at which dp and dm signals cross while transmitting in HS mode.		RW	
		2'b11	Default		
		2'b10	+15mv		
		2'b01	-15mv		
		2'b00	reserved		
3:0	TXVREFTUNE	These bits control HS DC voltage level adjustment.		RW	
		4'b1111	+12.5%		
		4'b1110	+11.25%		
		4'b1101	+10%		
		4'b1100	+8.75%		
		4'b1011	+7.5%		

		4'b1010	+6.255		
		4'b1001	+5%		
		4'b1000	+3.75%		
		4'b0111	+2.5%		
		4'b0110	+1.25%		
		4'b0101	Default		
		4'b0100	-1.25%		
		4'b0011	-2.5%		
		4'b0010	-3.75%		
		4'b0001	-5%		
		4'b0000	-6.25%		

11.1.2.22 USB Reset Detect Timer Register

It is initialized to 0x2000096

	USBRDT																														0x10000040
Bit	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
	Reserved					HB_MASK	VBFIL_LD_EN	IDDIG_EN	IDDIG_REG	USBRDT																							
RST	?	?	?	?	?	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0

Bits	Name	Description	RW
31:27	Reserved	Writing has no effect, read as zero.	R
26	HB_MASK	Halfword/Byte transfer support mask. 0: enable 1: mask	RW
25	VBFIL_LD_EN	VBUS filter data load enable.	RW
24	IDDIG_EN	This bit indicates using IDDIG_REG to control "iddig" signal.	RW
23	IDDIG_REG	This bit controls "iddig" when IDDIG_REG_EN = 1'b1.	RW
22:0	USBRDT	These bits control USB reset detect time.	RW

11.1.2.23 USB VBUS Jitter Filter Register

	USBVBFIL																														0x10000044
Bit	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
	IDDIGFIL																USBVBFIL															
RST	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:16	IDDIGFIL	These bits controls iddig jitter filter time.	RW
15:0	USBVBFIL	These bits controls VBUS jitter filter time.	RW

11.1.2.24 USB Parameter Control Register1

The USBPCR1 is a 32-bit read/write register that allows software to control OTG and OHCI PHY some functions. It is initialized to 0x8dc23360.

	USBPCR1																															0x10000048
Bit	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
	BVLD_REG	Reserved			REFCLKSEL		REFCLKDIV		Reserved		PORT0_RST	Reserved	WORD_IF0	Reserved																			
RST	1	?	?	?	1	1	0	1	?	?	0	?	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31	BVLD_REG	This bit is used to set "bvalid"(VBUS above B-device session threshold) signal.	R
30:28	Reserved	Writing has no effect, read as zero.	R
27:26	REFCLKSEL	This bus selects the OTG_PHY reference clock source 11, 10: The PLL uses CLKCORE as reference 01: The XO block uses an external, 2.5v clock supplied on the XO pin 00: The XO block uses the clock from a crystal.	RW
25:24	REFCLKDIV	This bus selects the OTG_PHY reference clock frequency 11: Reserved 10: 48MHz 01: 24MHz 00: 12MHz	RW
23:22	Reserved	Writing has no effect, read as zero.	R
21	PORT_RST	OTG PHY Port reset: 1: The transmit and receive finite state machines are reset, and the linestate logic combinatorially reflects the state of the single-ended receivers 0: The transmit and receive finite machines are operational, and the linestate inputs becomes sequential after 11 PHYCLOCK cycles.	RW
20	Reserved	Writing has no effect, read as zero.	R
19	WORD_IF	This bit selects utmi data bus width of otg 1: 16bit/30M	RW

		0: 8bit/60M	
18:0	Reserved	Writing has no effect, read as zero.	R

11.1.2.25 APLL Control Register

The APLL Control Register (CPAPCR) is a 32-bit read/write register, which controls PLL multiplier, on/off state and stabilize time. It is initialized to 0xa7000501 only by any reset. Only word access can be used on CPAPCR

	CPAPCR																													0x10000010
Bit	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
	BS	PLLM							Reserved	PLLN					PLLOD		LOCK0	Reserved				ON	PLLBP	PLLEN	PLLST							
RST	?	?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	Name	Description	RW
31	BS	0: low band 1: high band	RW
30:24	PLLM	the PLL feedback 7-bit divider.	RW
23	Reserved	Writing has no effect, read as zero.	R
22:18	PLLN	the PLL input 5-bit divider.	RW
17:16	PL OD	00: divide by 1 01: divide by 2 10: divide by 4 11: divide by 8	RW
15	LOCK0	0: the PLL output is not stable 1: the PLL output is stable Software should clear this bit to 0, when this bit equal to 1, it indicates that PLL hadn't stable previously , it is only used to debug.	R
14:11	Reserved	Writing has no effect, read as zero.	R
10	PLLON	PLL Stabilize Flag. 0: PLL is off or not stable 1: PLL is on and stable	R
9	PLLBP	PLL Bypass. 0: PLL output 1: EXCLK output **Don't change the bit. it is only used to debug.**	RW
8	PLLEN	PLL Enable.	RW
7:0	PLLST	PLL Stabilize Time. Specifies the PLL stabilize time by unit of RTCCLK (approximate 32kHz) cycles. It is used when change PLL multiplier or change PLL from off to on. It is initialized to H'20.	RW

11.1.2.26 MPLL Control Register

The MPLL Control Register (CPMPCR) is a 32-bit read/write register, which controls PLL multiplier, on/off state and stabilize time. It is initialized to 0xa7000081 only by any reset. Only word access can be used on CPMPCR.

	CPMPCR																0x10000014													
Bit	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
	BS	PLLM							Reserved	PLLN					PLLOD		Reserved								PLLEN	PLLBP	Reserved				LOCK	PLL1ON
RST	?	?	?	?	?	?	?	?	0	?	?	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bits	Name	Description	RW
31	BS	0: low band 1: high band	RW
30:24	PLLM	the PLL1 feedback 7-bit divider.	RW
23	Reserved	Writing has no effect, read as zero.	R
22:18	PLLN	the PLL1 input 5-bit divider.	RW
17:16	PLLOD	00: divide by 1 01: divide by 2 10: divide by 4 11: divide by 8	RW
15:8	Reserved	Writing has no effect, read as zero.	R
7	PLLEN	PLL Enable.	RW
6	PLLBP	PLL Bypass. 0: PLL output 1: EXCLK output **Don't change the bit. it is only used to debug.**	RW
5:1	Reserved	Writing has no effect, read as zero.	R
1	LOCK	0: the PLL output is not stable 1: the PLL output is stable Software should clear this bit to 0, when this bit equal to 1, it indicates that PLL hadn't stable previously , it is only used to debug.	R
0	PLLON	0: PLL1 doesn't enter on state 1: PLL1 is in on state	R

11.1.3 PLL Operation

The PLL developed as a macro cell for clock generator. It can generate a stable high-speed clock from a slower clock signal. The output frequency is adjustable and can be up to 1000MHz. The PLL

integrates a phase frequency detector (PFD), a low pass filter (LPF), a voltage controlled oscillator (VCO) and other associated support circuitry. All fundamental building blocks as well as fully programmable dividers are integrated on the core. It is useful for clock multiplication of stable crystal oscillator sources and for de-skew clock signals.
The PLL block diagram is shown in following figure:

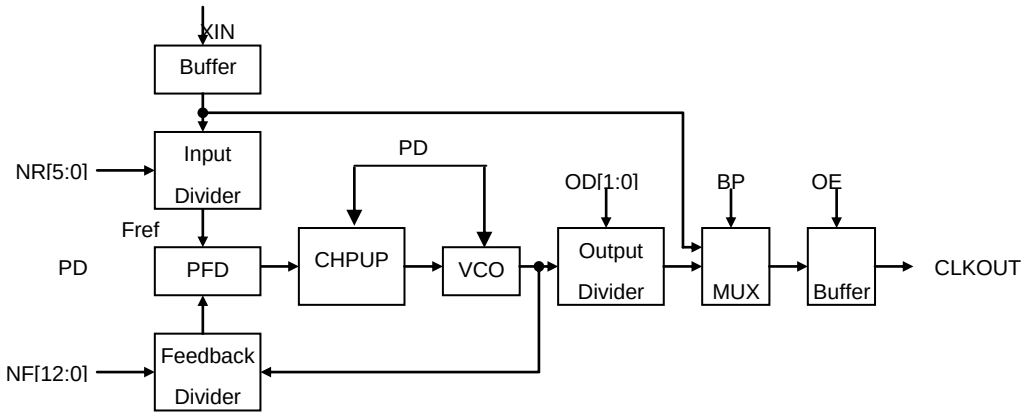


Figure 11-1 Block Diagram of PLL

11.1.3.1 PLL Configuration

PLL Divider Value Setting

There are 3 divider values (N, M and NO) to set the PLL output clock frequency CLKOUT:

1 Input Divider Value N.

N = PLLN of CPPCR

2 Feedback Divider Value M.

M = PLLM of CPPCR

3 Output Divider Value NO.

Output Divider Setting (OD)	Output Divider Value (NO)
0	1
1	2
2	4
3	8

4 The PLL output frequency, FOUT, is determined by the ratio set between the value set in the input divider and the feedback divider. PLL output frequency FOUT is calculated from the following equations:

NF = 1+ M0 + M1*2 + M2*4 + M3*8 + M4*16 + M5*32 + M6*64

NR = 1+ N0*1 + N1*2 + N2*4 + N3*8 + N4*16

NO = $2^{od0+2od1}$

FREF = FIN / NR

FVCO = FOUT * NO

FOUT = FIN * NF / (NR*NO), where FREF is the comparison frequency for the PFD.

For proper operation in normal mode, the following constraints must be satisfied:

For high-band,

10 MHz ⩽ FREF ⩽ 50 MHz

500 MHz ⩽ FVCO ⩽ 1000 MHz

62.5 MHz ⩽ FOUT ⩽ 1000 MHz

For low-band:

10 MHz ⩽ FREF ⩽ 50 MHz

300 MHz ⩽ FVCO ⩽ 600 MHz

37.5 MHz ⩽ FOUT ⩽ 600 MHz

11.1.4 Main Clock Division Change Sequence

Main clock (CCLK, L2CLK, H2CLK, PCLK) frequencies can be changed separately or simultaneously by changing division ratio. Following conditions must be obeyed:

1 the frequency of CCLK must be 1,2,3 or 4 times of the frequency of L2CLK

2 the frequency of H2CLK must be 1 or 2 times of the frequency of PCLK

Don't violate this limitation, otherwise unpredictable error may occur.

Important Note: When cpu enters sleep mode, the apllen must be set to 1.

11.2 Power Manager

In the Low-Power mode, part or whole processor is halted. This will reduce power consumption. The Power Management Controller contains low-power mode control and reset sequence control.

11.2.1 Low-Power Modes and Function

The processor supports six low-power modes and function:

- NORMAL mode

  In Normal mode, all peripherals and the basic blocks including power management block, the CPU core, the bus controller, the memory controller, the interrupt controller, DMA, and the external master may operate completely. But, the clock to each peripheral, except the basic blocks, can be stopped selectively by software to reduce the power consumption.

- IDLE mode
  In IDLE mode, the clock to the CPU core is stopped except the bus controller, the memory controller, the interrupt controller, and the power management block. To exit the IDLE mode, the any interrupts should be activated.

- SLEEP mode
  In SLEEP mode, all clocks except RTC clock are disabled. PLL is disabled also. SLEEP mode is canceled by reset or interrupt. When SLEEP mode is canceled, PLL is restarted , the PLL needs clock stabilization time (PLL lock time). This PLL stabilization time is automatically inserted by the internal logic with lock time count register. and all clocks start operating after PLL stability time.

- CLOCK GATE function
  CLOCK GATE function is used to gate specified on-chip module when it is not used. Set specified CLKG0~31 bits in CLKGR will enter specified CLK gate function. CLOCK gate function is canceled by reset or clearing specified CLKGR0~40 to 0.

- Power down Mode
  In order to reduce power leakage, software may shut down power supply for AHB1 and GPS module. When system enters into SLEEP mode, the software may shut down power for J1 according to OPCR.PD bit.

11.2.2 Register Description

All PMC register 32bit access address is physical address.

Table 11-3 Power/Reset Management Controller Registers Configuration

Name	description	RW	Initial Value	Offset Address	Access Size
LCR	Low Power Control Register	RW	0x00000000	0x0004	32
PSWC0ST	Power Switch Chain0 Start Time	RW	0x00000000	0x0090	32
PSWC1ST	Power Switch Chain1 Start Time	RW	0x00000000	0x0094	32
PSWC2ST	Power Switch Chain2 Start Time	RW	0x00000000	0x0098	32
PSWC3ST	Power Switch Chain3 Start Time	RW	0x00000000	0x009c	32
SRBC	Soft Reset and Bus Control Register	RW	0x00000000	0x00C4	32
SLBC	Sleep Boot Control Register	RW	0x00000000	0x00C8	32
SLPC	Sleep PC Register	RW	0x????????	0x00CC	32
CLKGR	Clock Gate Register0	RW	0x1FFFFF80	0x 0020	32
OPCR	Oscillator and Power Control Register	RW	0x10801500	0x 0024	32
MESTSEL	asnyc handshake control register	RW	0x00000000	0x00EC	32

11.2.2.1 Low Power Control Register

The Low Power Control Register (LCR) is a 32-bit read/write register that controls low-power mode status. It is initialized to 0x00000000 by any reset.

	LCR																BASE+0x0004													
Bit	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
	Reserved												PST												Reserved						LPM	
RST	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

Bits	Name	Description	RW
31:20	Reserved		R
19:8	PST	Power stability Time. Specifies the Power stabilize time by unit of RTCCLK (approximate 32kHz) cycles.	RW
7:2	Reserved		
1:0	LPM	Low Power Mode. Specifies which low-power mode will be entered when SLEEP instruction is executed. Bit 1~0: **00**: IDLE mode will be entered when SLEEP instruction is executed **01**: SLEEP mode will be entered when SLEEP instruction is executed	RW

11.2.2.2 Power Switch Chain0 Start Time Register

	PSWC0ST																0x10000090													
Bit	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
	Reserved																PSWC0ST															
RST	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

11.2.2.3 Power Switch Chain1 Start Time Register

	PSWC1ST																0x10000094													
Bit	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
	Reserved																PSWC1ST															
RST	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

11.2.2.4 Power Switch Chain2 Start Time Register

	PSWC2ST																													0x10000098	
Bit	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
	Reserved																PSWC2ST															
RST	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

11.2.2.5 Power Switch Chain3 Start Time Register

	PSWC3ST																													0x1000009C	
Bit	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
	Reserved																PSWC3ST															
RST	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

NOTE: The Start Time by the unit of PCLK cycles.

11.2.2.6 Clock Gate Register

The Clock Gate Register (CLKGR) is a 32-bit read/write register that controls the CLOCK GATE function of peripherals. It is reset to 0x07FFFF10.

	CLKGR																													BASE+0x0020	
Bit	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
	CLKGR																															
RST	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0

Bits	Name	Description	RW
31:0	CLKGR (Note1)	Clock gate Bits. Controls the clock supplies to some peripherals. If set, clock supplies to associated devices are stopped, and registers of the device cannot be accessed also. The reset value of this register reflects the clock gate information of the relative modules. RST 0: After reset period, the clock is not stopped. RST 1: After reset period, the clock is stopped.	RW

Bit	Module	Description

		31	DDR	If set, gate the clock.
		30	CPU	If set, not stop cpu when power down
		29	AHB0	If set, gate the clock.
		28	APB0	If set, gate the clock.
		27	RTC	If set, gate the clock.
		26	PCM	If set, gate the clock.
		25	MAC	If set, gate the clock.
		24	AES	If set, gate the clock.
		23	LCD	If set, gate the clock.
		22	CIM	If set, gate the clock.
		21	PDMA	If set, gate the clock.
		20	OST	If set, gate the clock.
		19	SSI	If set, gate the clock.
		18	TCU	If set, gate the clock.
		17	DMIC	If set, gate the clock.
		16	UART2	If set, gate the clock.
		15	UART1	If set, gate the clock.
		14	UART0	If set, gate the clock.
		13	SADC	Reserved
		12	JPEG	If set, gate the clock.
		11	AIC	If set, gate the clock.
		10	SMB3	Reserved
		9	SMB2	If set, gate the clock.
		8	SMB1	If set, gate the clock.
		7	SMB0(I2C0)	If set, gate the clock.
		6	SCC	If set, gate the clock. (Note2)
		5	MSC1	If set, gate the clock.
		4	MSC0	If set, gate the clock.
		3	OTG	If set, gate the clock.
		2	SFC	If set, gate the clock.
		1	EFUSE	If set, gate the clock.
		0	NEMC	If set, gate the clock.

Note1: you can use XXCDR.XX_STOP, CLKGR.XX or SRBC.XX_STP to stop XX module's clk, and when CLKGR.XX or SRBC.XX_STP is set high , CPU can turn into sleep mode successfully, but if only set XXCDR.XX_STOP high, CPU may not turn into sleep mode, because XX module can not give out stop_ack when clk is stopped; CPM will not wait for XX module's stop_ack if you stop it's clk by set CLKGR.XX.

Note2: Scc's clk is 1/8 of exclk.

11.2.2.7 CPM MEST SEL Register

	mestsel																													BASE+0x00EC	
Bit	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
	Reserved																							TST8	TST7	Reserved		TST4	TST3	Reserved	TST1	TST0
RST	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

Bits	Name	Description	RW
31:9	Reserved	Write has no effect,read as 0	R
8	TST8	AHB0 and APB0	RW
7	TST7	AHB2 and DDR	RW
6:5	Reserved	WR , but useless	RW
4	TST4	LCD and DDR	RW
3	TST3	CPU and DDR	RW
2	Reseved	WR , but useless	RW
1	TST1	CPU and AHB0	RW
0	TST0	CPU and AHB2	RW

11.2.2.8 Soft Reset and Bus Control Register (SRBC)

The Soft Reset and Bus Control Register is a 32-bit read/write register that control some module reset and bus transmission. It is initialized to 0x0000000 0by reset.Remember MAC's soft reset is in MPHYC.SOFT_RST.

	SRBC																													BASE+0x00C4	
Bit	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
	JPEG_SR	JPEG_STP	JPEG_ACK	Reserved			LCD_SR	LCD_STP	LCD_ACK	Reserved	CIM_STP	CIM_ACK	Reserved				CPU_STP	CPU_ACK	Reserved	OTG_SR	Reserved			AHB2_STP	AHB2_ACK	DDR_STP	DDR0_ACK	Reserved				
RST	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

Bits	Name	Description	RW
31	JPEG _SR	0: JPEG does not enter soft reset mode 1: JPEG enters soft reset mode	RW
30	JPEG_STP	Request for JPEG to Stop bus transfer.	RW
29	JPEG_ACK	JPEG Stop Ack.	R
28:26	Reserved	Write has no effect, Read as 0	RW
25	LCD_SR	0: LCD does not enter soft reset mode 1: LCD enters soft reset mode	RW

24	LCD_STP	Request for LCD to Stop bus transfer.	RW
23	LCD_ACK	LCD Stop Ack.	R
22	Reserved	Write has no effect, Read as 0	R
21	CIM_STP	Request for ISP to Stop bus transfer.	RW
20	CIM_ACK	ISP Stop Ack.	R
19:16	Reserved	Write has no effect, Read as 0	
15	CPU_STP	Request for CPU to Stop bus transfer.	RW
14	CPU_ACK	CPU Stop Ack.	R
13	Reserved	Write has no effect, Read as 0	R
12	OTG_SR	OTG soft reset	RW
11:9	Reserved	Write has no effect, Read as 0	R
8	AHB2_STP	Request for AHB2 to Stop bus transfer. Note: Do not chage the value of this bit,it's only used for debug	RW
7	AHB2_ACK	AHB2 Stop Ack.	R
6	DDR_STP	Request for DDR to Stop bus transfer.	RW
5	DDR_ACK	DDR Stop Ack.	R
4:0	Reserved	Write has no effect, Read as 0	R

11.2.2.9 Enable RNG Register (ERNG)

The Enable RNG Register is a 32-bit read/write register that control whether enable RNG register. It is initialized to 0x00000000 by reset.

	ERNG																BASE+0x00D8														
Bit	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
	ready	Reserved																														ERNG
RST	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

Bits	Name	Description	RW
31	ready	Data ready When software read RNG register, hardware auto clear ready to 0.	RC W
30:1	Reserved	Write has no effect, Read as 0	R
0	ERNG	0: not enable 1: enable	RW

11.2.2.10 RNG Register (RNG)

The RNG Register is a 32-bit read only registers.

	RNG																														BASE+0x00DC											
Bit	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										
	RNG																																								
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?										

Bits	Name	Description	RW
31:0	RNG	Random number output, read-only.	R

11.2.2.11 Sleep Boot Control Register (SLBC)

The Sleep Boot Control Register is a 32-bit read/write register that control sleep boot mode.It is initialized to 0x0000000 0by reset.

	SLBC																														0x100000C8											
Bit	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										
	Reserved																															SLBC										
RST	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										

Bits	Name	Description	RW
31:1	Reserved		R
0	SLBC	0: Sleep Mode Wake up Boot Process is same with any reset. 1: Sleep Mode Wake up Boot Process is jumped to SLPC, don't care Boot_Sel. The bit is only valid When shut down P0 in Sleep Mode	RW

11.2.2.12 Sleep PC Register (SLPC)

The Sleep PC Register is a 32-bit read/write register that control sleep mode jump address.

	SLPC																														0x100000CC											
Bit	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										
	SLPC																																								
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?										

Bits	Name	Description	RW
31:0	SLPC	When SLBC is 1, sleep boot is jumped to SLPC, not true boot as any	RW

		reset. The register is only valid When shut down P0 in Sleep Mode and SLBC is set high.	

11.2.2.13 Oscillator and Power Control Register (OPCR)

The Oscillator and Power Control Register is a 32-bit read/write register that specifies some special controls to oscillator and analog block. It is initialized to 0x00801500 by reset.

	OPCR																BASE+0x0024													
Bit	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
	IDLE_DIS	Mask_int_req	MASK_VPU	Gate_sclka_bus	Reserved		L2C_PD	REQ_MODE	Gate_usbphyclk	DIS_STOP_MUX	Reserved		O1ST												SPENDN0	SPENDN1	CPU_MODE	O1SE	PD	ERCS	BUS_MODE	Reserved
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	IDLE_DIS	0: when CPU enters idle mode, CPU clock is stopped 1: When CPU enters idle mode, CPU clock is not stopped	RW
30	MASK_INT	0: interrupt not mask 1: interrupt masked When enters sleep/idle mode, cpu should set MASK_INT 1, hardware will clear.	RW
29	MASK_VPU	Don't change this bit , it is only used to debug	RW
28	GATE_SCLK A_BUS	Gate SCLK_A for AHB0 AHB2 DDR	RW
27:26	Reserved	Write has no effect, read as 0	RW
25	L2C_PD	L2CC power down enable	R
24	REQ_MODE	0: Before go into sleep mode, CPM send stop req signal to relatvie modules in SRBC 1: Before go into sleep mode, CPM doesn't send stop req signal to those modules When DDR enters retention mode, software should be set the bit to 1	RW
23	gate_usbphy _clk	1:gate usb phy clk12m 0:not gate usb phy clk12m	
22	dis_stop_mu x	0: sel_mux clk of sel_src, sel_h0pp, sel_h2pll, sel_cpll, DDRCDR.DCS is stopped in sleep mode 1: those clk is not stopped in sleep mode	
21：20	Reserved	Write has no effect, read as 0	R

19:8	O1ST	EXCLK Oscillator Stabilize Time. This filed specifies the EXCLK oscillator stabilize time by unit of 16 RTCCLK periods (oscillator stable time O1ST × 16 / 32768) cycles. It is initialized to H'15.	RW
7	SPENDN0	force USB PHY to enter suspend mode when OTG is enable. 0: USB PHY has forced to entered SUSPEND mode 1: USB PHY hasn't forced to entered SUSPEND mode	RW
6	SPENDN1	force USB PHY to enter suspend mode when UHC is enable. 0: USB PHY has forced to entered SUSPEND mode 1: USB PHY hasn't forced to entered SUSPEND mode	RW
5	CPU_MODE		RW
4	O1SE	EXCLK Oscillator Sleep Mode Enable. This filed controls the state of the EXCLK oscillator in Sleep mode. 0: EXCLK oscillator is disabled in Sleep mode 1: EXCLK oscillator is enabled in Sleep mode	RW
3	PD	0: no power down cpu_core in sleep mode 1: power down cpu_core in sleep mode	RW
2	ERCS	EXCLK/512 clock and RTCLK clock selection. 0: select EXCLK/512 division ration clock 1: select RTCLK clock the clock only output to CPM INTC SSI TCU etc.	RW
1	BUS_MODE	The default value is 0 When the value is 1, the BUS access is accelerated.	RW
0	reserved	Write has no effect, Read as 0	R

Note: Before you want to power down cpu buy set PD_P0, PD_P1 or PD in OPCR, you need to reduce cpu's frequency as the same as L2C's freq; or else unpredictable error will accurs when CPU is waken up again.

11.2.3 IDLE Mode

In normal mode, when LPM bits in LCR are 0 and SLEEP instruction is executed, the processor enters idle mode. CPU is halted but its register contents are retained All critical application must be finished and peripherals must be configured to generate interrupts when they need CPU attention.

The procedure of entering sleep mode is shown blow:

1 Set LPM bits in LCR to 0.
2 Executes SLEEP instruction.
3 When current operation of CPU core has finished and CPU core is idle, CCLK supply to CPU core is stopped.

IDLE mode is exited by an interrupt (IRQ or on-chip devices) or a reset.

11.2.4 SLEEP Mode

In normal mode, when LPM bits in LCR is 1 and SLEEP instruction is executed, the processor enter SLEEP mode. CPU and on-chip devices are halted, except some wakeup-logic. PLL is shut off. Clock output from CKO pin is also stopped. SDRAM content is preserved by driving into self-refresh state. CPU registers and on-chip devices registers contents are retained.

Before enter SLEEP mode, software should ensure that all peripherals are not running. The procedure of entering SLEEP mode is shown blow:

1 Set LPM bit in LCR to 1.
2 Execute a SLEEP instruction.
3 When current access on system bus complete, the arbiter will not grant any following request. EMC will drive SDRAM from auto-refresh mode to self-refresh mode.
4 When system bus is idle state and SDRAM is self-refresh mode, internal clock supplies are stopped.
5 SLEEP mode can be exited by an interrupt (IRQ or on-chip devices), WDT reset or a poweron reset via the RESETP pin.

11.3 Reset Control Module

11.3.1 Register Description

All RCM register 32bit access address is physical address.

Name	description	RW	Initial Value	Address	Access Size
RSR	Reset Status Register	RW	0x????????	0x0008	32

11.3.1.1 Reset Status Register (RSR)

The Reset Status Register (RSR) is a 32-bit read/write register which records last cause of reset. Each RSR bit is set by a different source of reset. Please refer to Reset Sequence Control for reset sources description.

	RSR																BASE+0x0008															
Bit	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
	Reserved																												HR	POR	WR	PR
RST	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	?	?	?

Bits	Name	Description	RW
31:4	Reserved	Writing has no effect, read as zero.	R
3	HR	Hibernate Reset. Software can only write 0,Write 1 will be ignored	RW
2	P0R	P0 power up Reset. It indicates that P0 has been shut down, now it has been power up. When P0 reset is detected, P0R is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 will be ignored. 0: P0 reset has not occurred since the last time the software clears this bit 1: P0 reset has occurred since the last time the software clears this bit	
1	WR	WDT Reset. When a WDT reset is detected, WR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 will be ignored. 0: WDT reset has not occurred since the last time the software clears this bit 1: WDT reset has occurred since the last time the software clears this bit	RW
0	PR	Power On Reset. When a power-on reset via PRESET pin is detected, PR is set and remains set until software clears it or another reset occurs. This bit can only be written with 0. Write with 1 is ignored. 0: Power on reset has not occurred since the last time the software clears this bit 1: Power on reset has occurred since the last time the software clears this bit	RW

11.3.2 Power On Reset

Power on reset is generated when PRESET pin is driven to low. Internal reset is asserted immediately. All pins return to their reset states. The Power on reset is extended to 40MS.

PRESET pin must be held low until power stabilizes and the EXCLK oscillator stabilize. CPU and peripherals are clocked by EXCLK oscillator output directly. PLL is reset to off state. All internal modules are initialized to their predefined reset states.

11.3.3 WDT Reset

WDT reset is generated when WDT overflow. Internal reset is asserted within two RTCCLK cycles. All pins return to their reset states.

Then WDT reset source is cleared because of internal reset. The internal reset is asserted for about 10 milliseconds. CPU and peripherals are clocked by EXCLK oscillator output directly. PLL is reset to off state.

12 Timer/Counter Unit

12.1 Overview

The TCU (Timer/Counter with PWM output) contains 8 channels of 16-bit programmable timers. They can be used as Timer or PWM. Only ch0~ch4 can export PWM

TCU has the following features:

- There are two modes of TCU for the eight channels
  - TCU1: Channel 0, 3,4, 5, 6,and 7
  - TCU2: Channel 1,2
- Six independent channels, each consisting of
  - Counter
  - Data register (FULL and HALF)
  - Control register
- Independent clock for each counter, selectable by software
  - PCLK, EXTAL and RTCCLK can be used as the clock for counter
  - The division ratio of the clock can be set to 1, 4, 16, 64, 256 and 1024 by software
- FULL interrupt and HALF interrupt can be generated for each channel using the compare data registers
  - Timer 0-4 can be used as PWM (Set the initial signal level)
  - Timer 0,3-7 can be used as a counter to count external signal (like trackball)
  - Timer 5 has separated interrupt
  - Timer 0-4 and timer 6-7 has one interrupt in common
  - OST uses interrupt 0, Timer 5 uses interrupt 1, and Timer 0-4/ 6-7 uses interrupt 2
- The difference between TCU1 and TCU2
  - TCU1: It cannot work in sleep mode, but operated easily
  - TCU2: It can work in sleep mode, but operated more complicated than TCU1

12.2 Pin Description

Table 12-1 PWM Pins Description

Name	I/O	Description
PWM [4:0]	Output	PWM channel output signals.

12.3 Register Description

Following chapter will descript the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute
   R - Read only
   W - Write only
   RW - read and write
   RCW - read and write, but clear to 0 by read
   RSW - read and write, but set to 1 by read
   RWC - read and write, clear to 0 by write 1, write 0 has no effect
   RWS - read and write, set to 1 by write 1, write 0 has no effect
   RC - read only, and clear to 0 by read
   RS - read only, and set to 1 by read
   SPEC - special access method, relate to its description
3. Reset Value
   1 - reset to 1
   0 - reset to 0
   ? - value unknown after reset

Table 12-2 Registers Memory Map-Address Base

Name	Base	Description
TCU	0x10002000	Address base of TCU

Table 12-3 TCU Registers Configuration

Name	Description	RW	Reset Value	Address offset	Access Size
TSTR	Timer Status Register	R	0x00000000	0x0F0	32
TSTSR	Timer Status Set Register	W	0x????????	0x0F4	32
TSTCR	Timer Status Clear Register	W	0x????????	0x0F8	32
TSR	Timer STOP Register	R	0x00000000	0x01C	32
TSSR	Timer STOP Set Register	W	0x00000000	0x02C	32
TSCR	Timer STOP Clear Register	W	0x0000	0x03C	32
TER	Timer Counter Enable Register	R	0x0000	0x010	16
TESR	Timer Counter Enable Set Register	W	0x????	0x014	16
TECR	Timer Counter Enable Clear Register	W	0x????	0x018	16
TFR	Timer Flag Register	R	0x003F003F	0x020	32
TFSR	Timer Flag Set Register	W	0x????????	0x024	32
TFCR	Timer Flag Clear Register	W	0x????????	0x028	32

TMR	Timer Mask Register	R	0x00000000	0x030	32
TMSR	Timer Mask Set Register	W	0x????????	0x034	32
TMCR	Timer Mask Clear Register	W	0x????????	0x038	32
TDFR0	Timer Data FULL Register 0	RW	0x????	0x040	16
TDHR0	Timer Data HALF Register 0	RW	0x????	0x044	16
TCNT0	Timer Counter 0	RW	0x????	0x048	16
TCSR0	Timer Control Register 0	RW	0x0000	0x04C	16
TDFR1	Timer Data FULL Register 1	RW	0x????	0x050	16
TDHR1	Timer Data HALF Register 1	RW	0x????	0x054	16
TCNT1	Timer Counter 1	RW	0x????	0x058	16
TCSR1	Timer Control Register 1	RW	0x0000	0x05C	16
TDFR2	Timer Data FULL Register 2	RW	0x????	0x060	16
TDHR2	Timer Data HALF Register 2	RW	0x????	0x064	16
TCNT2	Timer Counter 2	RW	0x????	0x068	16
TCSR2	Timer Control Register 2	RW	0x0000	0x06C	16
TDFR3	Timer Data FULL Register 3	RW	0x????	0x070	16
TDHR3	Timer Data HALF Register 3	RW	0x????	0x074	16
TCNT3	Timer Counter 3	RW	0x????	0x078	16
TCSR3	Timer Control Register 3	RW	0x0000	0x07C	16
TDFR4	Timer Data FULL Register 4	RW	0x????	0x080	16
TDHR4	Timer Data HALF Register 4	RW	0x????	0x084	16
TCNT4	Timer Counter 4	RW	0x????	0x088	16
TCSR4	Timer Control Register 4	RW	0x0000	0x08C	16
TDFR5	Timer Data FULL Register 5	RW	0x????	0x090	16
TDHR5	Timer Data HALF Register 5	RW	0x????	0x094	16
TCNT5	Timer Counter 5	RW	0x????	0x098	16
TCSR5	Timer Control Register 5	RW	0x0000	0x09C	16
TDFR6	Timer Data FULL Register 6	RW	0x????	0x0A0	16
TDHR6	Timer Data HALF Register 6	RW	0x????	0x0A4	16
TCNT6	Timer Counter 6	RW	0x????	0x0A8	16
TCSR6	Timer Control Register 6	RW	0x0000	0x0AC	16
TDFR7	Timer Data FULL Register 7	RW	0x????	0x0B0	16
TDHR7	Timer Data HALF Register 7	RW	0x????	0x0B4	16
TCNT7	Timer Counter 7	RW	0x????	0x0B8	16
TCSR7	Timer Control Register 7	RW	0x0000	0x0BC	16

12.3.1 Timer Control Register (TCSR)

The TCSR is a 16-bit read/write register. It contains the control bits for each channel. It is initialized to 0x00 by any reset.

	TCSR0, TCSR1, TCSR2, TCSR3, TCSR4, TCSR5 TCSR6, TCSR7															BASE+0x04C, 0x05C, 0x06C, 0x07C, 0x08C, 0x09C, 0x0AC, 0x0BC													
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																Reserved				BYPASS	CLRZ	SD	INITL	PWM_EN	PWM_IN_EN	PRESCALE			EXT_EN	RTC_EN	PCK_EN
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:11	Reserved	Writing has no effect, read as zero.	R
11	BYPASS	PWM bypass mode. 1: If PCK_EN = 1, this channel output PIXCLK; If RTC_EN = 1, this channel output RTCCLK; If EXT_EN = 1, this channel output EXTAL; Only one of those XXX_EN is permit available during one time. 0: This BYPASS function disable. *Only when you want to let this PWM channel output some special clock (PCLK, RTCLK and EXTAL clock), you can set this register to 1. Otherwise keep it to 0. When you want to use BYPASS function, not forget offer clock supplies of relate channel (relate to register TSR, TSSR, TSCR).(the current version can not bypass pclk,and when bypass a clk,the relative timer will be stopped to reduce power consumption.)	RW
10	CLRZ	Clear counter to 0. It is only used in TCU2 mode. Writing 1 to this bit will clear the counter to 0. When the counter is finished setting to 0, it will be cleared by hardware. Writing 0 to this bit will be ignored.	RW
9	SD	Shut Down (SD) the PWM output. It is only used in TCU1 mode. 0: Graceful shutdown 1: Abrupt shutdown Graceful shutdown: The output level for PWM output will keep the level only after the comparison match of FULL.(when write register to stop the timer,the timer will continue run utill full match) Abrupt shutdown: The output level for PWM output will keep the level as soon as software write to stop the counter.(the conter will stop as soon as softerware write to stop the counter).	RW
8	INITL	Selects an initial output level for PWM output. 0: Low 1: High	RW
7	PWM_EN	PWM output pin control bit.	RW

<table>
<tr><td></td><td></td><td>0: PWM pin output disable, and the PWM pin will be set to the initial level according to INITL
1: PWM pin output enable(only channel0~channel4 are available currently)</td><td></td></tr>
<tr><td>6</td><td>PWM_IN_EN</td><td>PWM input mode enable.
Set to 1 to enable this function.
In this function, PWM pin need to set GPIO as function0 to receive external signal, EXT_EN, RTC_EN, PCK_EN need to set 0. And TCNT became a counter to count this signal’s both edges. (This bit in TCSR1, 2 are Reserved, and onlye ch0,ch3 and ch4 are availabel currently).</td><td>RW</td></tr>
<tr><td>5:3</td><td>PRESCALE</td><td>These bits select the TCNT count clock frequency. Don’t change this field when the channel is running.
<table>
<tr><th>Bit 2</th><th>Bit1</th><th>Bit 0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Internal clock: CLK/1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Internal clock: CLK/4</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Internal clock: CLK/16</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Internal clock: CLK/64</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Internal clock: CLK/256</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Internal clock: CLK/1024</td></tr>
<tr><td colspan="3">110~111</td><td>Reserved</td></tr>
</table></td><td>RW</td></tr>
<tr><td>2</td><td>EXT_EN</td><td>Select EXTAL as the timer clock input.
0: Disable
1: Enable</td><td>RW</td></tr>
<tr><td>1</td><td>RTC_EN</td><td>Select RTCCLK as the timer clock input.
0: Disable
1: Enable</td><td>RW</td></tr>
<tr><td>0</td><td>PCK_EN</td><td>Select PCLK as the timer clock input.
0: Disable
1: Enable</td><td>RW</td></tr>
</table>

NOTE: The input clock of timer and the PCLK should keep to the rules as follows:

Input clock of timer: IN_CLK	**Clock generated from the frequency divider (PRESCALE): DIV_CLK**
PCK_EN == 0, RTC_EN == 1 and EXT_EN == 0 (IN_CLK = RTCCLK)	$f_{DIV\_CLK} < ½ f_{PCLK}$
PCK_EN == 0, RTC_EN == 0 and EXT_EN == 1 (IN_CLK = EXTAL)	$f_{DIV\_CLK} < ½ f_{PCLK}$
PCK_EN == 1, RTC_EN == 0 and EXT_EN == 0 (IN_CLK = PCLK)	ANY

12.3.2 Timer Data FULL Register (TDFR)

The comparison data FULL registers TDFR is used to store the data to be compared with the content of the up-counter TCNT. This register can be directly read and written. (Default: indeterminate) But it is not suggested changing when counter is working in TCU2 mode.

	TDFR0, TDFR1, TDFR2, TDFR3, TDFR4, TDFR5, TDFR6, TDFR7														BASE+0x040, 0x050, 0x060, 0x070, 0x080, 0x090, 0x0A0, 0x0B0														
Bit															15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															TDFR															
RST															?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

12.3.3 Timer Data HALF Register (TDHR)

The comparison data HALF registers TDHR is used to store the data to be compared with the content of the up-counter TCNT. This register can be directly read and written. (Default: indeterminate) But it is not suggested changing when counter is working in TCU2 mode.

	TDHR0, TDHR1, TDHR2, TDHR3, TDHR4, TDHR5, TDHR6, TDHR7														BASE+0x044, 0x054, 0x064, 0x074, 0x084, 0x094, 0x0A4, 0x0B4														
Bit															15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
															TDHR															
RST															?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

12.3.4 Timer Counter (TCNT)

TCNT is a 16-bit read/write register. The up-counter TCNT can be reset to 0 by software and counts up using the prescaler output clock. When TCNT count up to equal to TDFR, it will reset to 0 and continue to count up.

TCU1: The counter data can be read out at any time. The data can be written at any time. This makes it possible to change the interrupt and/or clock output cycles temporarily. (Default: indeterminate)

TCU2: The counter data can be read out at any time, but you should read TSTR.REALn to check whether the data is real data or not. The data can only be written before counter is started, and the counter clock is pclk. But it can be cleared to 0 by setting TCSR.CLRZ to 1, and if the counter is really cleared, TCSR.CLRZ will be set to 0 by hardware.

	TCNT0, TCNT1, TCNT2, TCNT3, TCNT4, TCNT5, TCNT6, TCNT7															BASE+0x048, 0x058, 0x068, 0x078, 0x088, 0x098, 0x0A8, 0x0B8															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																TCNT															
RST																?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

12.3.5 Timer Counter Enable Register (TER)

The TER is a 16-bit read-only register. It contains the counter enable control bits for each channel. It is initialized to 0x0000 by any reset. It can only be set by register TESR and TECR. Since the timer enable control bits are located in the same addresses, two or more timers can be started at the same time.

	TER															BASE+0x010															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																OSTEN	Reserved							TCEN 7	TCEN 6	TCEN 5	TCEN 4	TCEN 3	TCEN 2	TCEN 1	TCEN 0
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15	OSTEN	Enable the counter in OST. 0: Stop counting up 1: Begin counting up	R
14:8	Reserved	Writing has no effect, read as zero.	R
7	TCEN 7	Enable the counter in timer 7. 0: Stop counting up 1: Begin counting up	R
6	TCEN 6	Enable the counter in timer 6. 0: Stop counting up 1: Begin counting up	R
5	TCEN 5	Enable the counter in timer 5. 0: Stop counting up 1: Begin counting up	R
4	TCEN 4	Enable the counter in timer 4. 0: Stop counting up 1: Begin counting up	R
3	TCEN 3	Enable the counter in timer 3. 0: Stop counting up 1: Begin counting up	R
2	TCEN 2	Enable the counter in timer 2.	R

		0: Stop counting up 1: Begin counting up	
1	TCEN 1	Enable the counter in timer 1. 1: Begin counting up 0: Stop counting up	R
0	TCEN 0	Enable the counter in timer 0. 0: Stop counting up 1: Begin counting up	R

12.3.6 Timer Counter Enable Set Register (TESR)

The TCCSR is a 32-bit write-only register. It contains the counter enable set bits for each channel. Since the timer enable control set bits are located in the same addresses, two or more timers can be started at the same time.

	TESR																BASE+0x014													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	OSTST	Reserved							TCST 7	TCST 6	TCST 5	TCST 4	TCST 3	TCST 2	TCST 1	TCST 0
RST																	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
15	OSTST	Set OSTEN bit of TER. 0: Ignore 1: Set OSTEN bit to 1	W
14:8	Reserved	Writing has no effect, read as zero.	R
7	TCST 7	Set TCEN 7 bit of TER. 0: Ignore 1: Set TCEN 5 bit to 1	W
6	TCST 6	Set TCEN 6 bit of TER. 0: Ignore 1: Set TCEN 5 bit to 1	W
5	TCST 5	Set TCEN 5 bit of TER. 0: Ignore 1: Set TCEN 5 bit to 1	W
4	TCST 4	Set TCEN 4 bit of TER. 1: Set TCEN 4 bit to 1 0: Ignore	W
3	TCST 3	Set TCEN 3 bit of TER. 0: Ignore 1: Set TCEN 3 bit to 1	W
2	TCST 2	Set TCEN 2 bit of TER.	W

		1: Set TCEN 2 bit to 1 0: Ignore	
1	TCST 1	Set TCEN 1 bit of TER. 0: Ignore 1: Set TCEN 1 bit to 1	W
0	TCST 0	Set TCEN 0 bit of TER. 0: Ignore 1: Set TCEN 0 bit to 1	W

12.3.7 Timer Counter Enable Clear Register (TECR)

The TECR is a 32-bit write-only register. It contains the counter enable clear bits for each channel. Since the timer enable clear bits are located in the same addresses, two or more timers can be stop at the same time.

TECR																BASE+0x018															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																OSTCL	Reserved							TCCL 7	TCCL 6	TCCL 5	TCCL 4	TCCL 3	TCCL 2	TCCL 1	TCCL 0
RST																?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
15	OSTCL	Set OSTEN bit of TER. 0: Ignore 1: Set OSTEN 5 bit to 0	W
14:8	Reserved	Writing has no effect, read as zero.	R
7	TCCL 7	Set TCEN 7 bit of TER. 0: Ignore 1: Set TCEN 6 bit to 0	W
6	TCCL 6	Set TCEN 7 bit of TER. 0: Ignore 1: Set TCEN 6 bit to 0	W
5	TCCL 5	Set TCEN 5 bit of TER. 0: Ignore 1: Set TCEN 5 bit to 0	W
4	TCCL 4	Set TCEN 4 bit of TER. 1: Set TCEN 4 bit to 0 0: Ignore	W
3	TCCL 3	Set TCEN 3 bit of TER. 0: Ignore 1: Set TCEN 3 bit to 0	W
2	TCCL 2	Set TCEN 2 bit of TER.	W

		1: Set TCEN 2 bit to 0 0: Ignore	
1	TCCL 1	Set TCEN 1 bit of TER. 0: Ignore 1: Set TCEN 1 bit to 0	W
0	TCCL 0	Set TCEN 0 bit of TER. 0: Ignore 1: Set TCEN 0 bit to 0	W

12.3.8 Timer Flag Register (TFR)

The TFR is a 32-bit read-only register. It contains the comparison match flag bits for all the channels. It can also be set by register TFSR and TFCR. It is initialized to 0x00000000 by any reset.

	TFR																BASE+0x020													
Bit	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
	Reserved								HFLAG 7	HFLAG 6	HFLAG 5	HFLAG 4	HFLAG 3	HFLAG 2	HFLAG 1	HFLAG 0	OSTFLAG	Reserved							FFLAG 7	FFLAG 6	FFLAG 5	FFLAG 4	FFLAG 3	FFLAG 2	FFLAG 1	FFLAG 0
RST	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

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	HFLAG 7~0	HALF comparison match flag. (TCNT = TDHR) 0: Comparison not match 1: Comparison match	R
15	OSTFLAG	OST comparison match flag. (OSTCNT = OSTDR) 0: Comparison not match 1: Comparison match	R
14:8	Reserved	Writing has no effect, read as zero.	R
7:0	FFLAG 7~0	FULL comparison match flag. (TCNT = TDFR) 0: Comparison not match 1: Comparison match	R

12.3.9 Timer Flag Set Register (TFSR)

The TFSR is a 32-bit write-only register. It contains the comparison match flag set bits for all the channels.

	TFSR																BASE+0x024													
Bit	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
	Reserved								HFST 7	HFST 6	HFST 5	HFST 4	HFST 3	Reserved		HFST 0	OSTFST	Reserved							FFST 7	FFST 6	FFST 5	FFST 4	FFST 3	Reserved		FFST 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:19	HFST 7~3	Set HFLAG n bit of TFR. 0: Ignore 1: Set HFLAG n bit to 1	W
18:17	Reserved		
16	HFST0	Set HFLAG 0 bit of TFR. 0: Ignore 1: Set HFLAG n bit to 1	W
15	OSTFST	Set OSTFLAG n bit of TFR. 0: Ignore 1: Set OSTFLAG n bit to 1	W
14:8	Reserved	Writing has no effect, read as zero.	R
7:3	FFST 7~0	Set FFLAG n bit of TFR. 0: Ignore 1: Set FFLAG n bit to 1	W
2:1	Reserved	Writing has no effect, read as zero.	
0	FFST0	Set FFLAG 0 bit of TFR. 0: Ignore 1: Set FFLAG n bit to 1	

12.3.10 Timer Flag Clear Register (TFCR)

The TFCR is a 32-bit write-only register. It contains the comparison match flag clear bits for all the channels.

	TFCR																BASE+0x028													
Bit	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
	Reserved								HFCL 7	HFCL 6	HFCL 5	HFCL 4	HFCL 3	HFCL 2	HFCL 1	HFCL 0	OSTFCL	Reserved							FFCL 7	FFCL 6	FFCL 5	FFCL 4	FFCL 3	FFCL 2	FFCL 1	FFCL 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	HFCL 7~0	Set HFLAG n bit of TFR. 0: Ignore 1: Set FFLAG n bit to 0	W
15	OSTFCL	Set OSTFLAG n bit of TFR. 0: Ignore 1: Set OSTFLAG n bit to 0	W
14:8	Reserved	Writing has no effect, read as zero.	R
7:0	FFCL 7~0	Set FFLAG n bit of TFR. 0: Ignore 1: Set FFLAG n bit to 0	W

12.3.11 Timer Mast Register (TMR)

The TMR is a 32-bit read-only register. It contains the comparison match flag bits for all the channels. It is initialized to 0x003F003F by any reset. It can only be set by register TMSR and TMCR.

	TMR																BASE+0x030													
Bit	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
	Reserved								HMASK 7	HMASK 6	HMASK 5	HMASK 4	HMASK 3	HMASK 2	HMASK 1	HMASK 0	OSTMASK	Reserved							FMASK 7	FMASK 6	FMASK 5	FMASK 4	FMASK 3	FMASK 2	FMASK 1	FMASK 0
RST	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	HMASK 7~0	HALF comparison match interrupt mask. 0: Comparison match interrupt not mask 1: Comparison match interrupt mask	R
15	OSTMASK	OST comparison match interrupt mask. 0: Comparison match interrupt not mask 1: Comparison match interrupt mask	R
14:8	Reserved	Writing has no effect, read as zero.	R
7:0	FMASK 7~0	FULL comparison match interrupt mask. 0: Comparison match interrupt not mask 1: Comparison match interrupt mask	R

12.3.12 Timer Mask Set Register (TMSR)

The TMSR is a 32-bit write-only register. It contains the comparison match flag set bits for all the channels.

	TMSR																BASE+0x034													
Bit	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
	Reserved								HMST 7	HMST 6	HMST 5	HMST 4	HMST 3	HMST 2	HMST 1	HMST 0	OSTMST	Reserved							FMST 7	FMST 6	FMST 5	FMST 4	FMST 3	FMST 2	FMST 1	FMST 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	HMST 7~0	Set HMASK n bit of TMR. 0: Ignore 1: Set HMASK n bit to 1	W
15	OSTMST	Set OSTMASK n bit of TMR. 0: Ignore 1: Set OSTMASK n bit to 1	W
14:8	Reserved	Writing has no effect, read as zero.	R
7:0	FMST 7~0	Set FMASK n bit of TMR. 0: Ignore 1: Set FMASK n bit to 1	W

12.3.13 Timer Mask Clear Register (TMCR)

The TMCR is a 32-bit write-only register. It contains the comparison match flag clear bits for all the channels.

	TMCR																BASE+0x038													
Bit	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
	Reserved								HMCL 7	HMCL 6	HMCL 5	HMCL 4	HMCL 3	HMCL 2	HMCL 1	HMCL 0	OSTMCL	Reserved							FMCL 7	FMCL 6	FMCL 5	FMCL 4	FMCL 3	FMCL 2	FMCL 1	FMCL 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	HMCL 7~0	Set HMASK n bit of TMR. 0: Ignore 1: Set HMASK n bit to 0	W

15	OSTMCL	Set OSTMASK n bit of TMR. 0: Ignore 1: Set OSTMASK n bit to 0	W
14:8	Reserved	Writing has no effect, read as zero.	R
7:0	FMCL 7~0	Set FMASK n bit of TMR. 0: Ignore 1: Set FMASK n bit to 0	W

12.3.14 Timer Stop Register (TSR)

The TSR is a 32-bit read-only register. It contains the timer stop control bits for each channel, WDT and OST. It is initialized to 0x00000000 by any reset. It can only be set by register TSSR and TSCR. If set, clock supplies to timer n / WDT / OST is stopped, and registers of the timer / WDT / OST cannot be accessed also.

	TSR																BASE+0x01C													
Bit	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
	Reserved															WDTS	OSTS	Reserved							STOP 7	STOP 6	STOP 5	STOP 4	STOP 3	STOP 2	STOP 1	STOP 0
RST	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

Bits	Name	Description	RW
31:17	Reserved	Writing has no effect, read as zero.	R
16	WDTS	0: The clock supplies to WDT is supplied 1: The clock supplies to WDT is stopped	R
15	OSTS	0: The clock supplies to OST is supplied 1: The clock supplies to OST is stopped	R
14:8	Reserved	Writing has no effect, read as zero.	R
7	STOP 7	0: The clock supplies to timer 7 is supplied 1: The clock supplies to timer 7 is stopped	R
6	STOP 6	0: The clock supplies to timer 6 is supplied 1: The clock supplies to timer 6 is stopped	R
5	STOP 5	0: The clock supplies to timer 5 is supplied 1: The clock supplies to timer 5 is stopped	R
4	STOP 4	0: The clock supplies to timer 4 is supplied 1: The clock supplies to timer 4 is stopped	R
3	STOP 3	0: The clock supplies to timer 3 is supplied 1: The clock supplies to timer 3 is stopped	R
2	STOP 2	0: The clock supplies to timer 2 is supplied 1: The clock supplies to timer 2 is stopped	R

1	STOP 1	0: The clock supplies to timer 1 is supplied 1: The clock supplies to timer 1 is stopped	R
0	STOP 0	0: The clock supplies to timer 0 is supplied 1: The clock supplies to timer 0 is stopped	R

12.3.15 Timer Stop Set Register (TSSR)

The TCSR is an 32-bit write-only register. It contains the timer stop set bits for each channel, WDT and OST. Since the timer stop control set bits are located in the same addresses, two or more timers can be started at the same time.

	TSSR																BASE+0x02C													
Bit	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
	Reserved															WDTSS	OSTSS	Reserved							STPS 7	STPS 6	STPS 5	STPS 4	STPS 3	STPS 2	STPS 1	STPS 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:17	Reserved	Writing has no effect, read as zero.	R
16	WDTSS	Set WDTS bit of TSR. 0: Ignore 1: Set WDTS bit to 1	W
15	OSTSS	Set OSTS bit of TSR. 0: Ignore 1: Set OSTS bit to 1	W
14:8	Reserved	Writing has no effect, read as zero.	R
7	STPS 7	Set STOP 7 bit of TSR. 0: Ignore 1: Set STOP 7 bit to 1	W
6	STPS 6	Set STOP 6 bit of TSR. 0: Ignore 1: Set STOP 6 bit to 1	W
5	STPS 5	Set STOP 5 bit of TSR. 0: Ignore 1: Set STOP 5 bit to 1	W
4	STPS 4	Set STOP 4 bit of TSR. 1: Set STOP 4 bit to 1 0: Ignore	W
3	STPS 3	Set STOP 3 bit of TSR. 0: Ignore 1: Set STOP 3 bit to 1	W

2	STPS 2	Set STOP 2 bit of TSR. 0: Ignore 1: Set STOP 2 bit to 1	W
1	STPS 1	Set STOP 1 bit of SR. 0: Ignore 1: Set STOP 1 bit to 1	W
0	STPS 0	Set STOP 0 bit of TSR. 0: Ignore 1: Set STOP 0 bit to 1	W

12.3.16 Timer Stop Clear Register (TSCR)

The TSCR is an 32-bit write-only register. It contains the timer stop clear bits for each channel, WDT and OST. Since the timer stop clear bits are located in the same addresses, two or more timers can be stop at the same time.

	TSCR																BASE+0x03C													
Bit	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
	Reserved															WDTSC	OSTSC	Reserved							STPC 7	STPC 6	STPC 5	STPC 4	STPC 3	STPC 2	STPC 1	STPC 0
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:17	Reserved	Writing has no effect, read as zero.	R
16	WDTSC	Set WDTS bit of TSR. 0: Ignore 1: Set WDTS bit to 0	W
15	OSTSC	Set OSTS bit of TSR. 0: Ignore 1: Set OSTS bit to 0	W
14:8	Reserved	Writing has no effect, read as zero.	R
7	STPC 7	Set STOP 7 bit of TSR. 0: Ignore 1: Set STOP 7 bit to 0	W
6	STPC 6	Set STOP 6 bit of TSR. 0: Ignore 1: Set STOP 6 bit to 0	W
5	STPC 5	Set STOP 5 bit of TSR. 0: Ignore 1: Set STOP 5 bit to 0	W
4	STPC 4	Set STOP 4 bit of TSR. 0: Ignore	W

		1: Set STOP 4 bit to 0	
3	STPC 3	Set STOP 3 bit of TSR. 0: Ignore 1: Set STOP 3 bit to 0	W
2	STPC 2	Set STOP 2 bit of TSR. 0: Ignore 1: Set STOP 2 bit to 0	W
1	STPC 1	Set STOP 1 bit of TSR. 0: Ignore 1: Set STOP 1 bit to 0	W
0	STPC 0	Set STOP 0 bit of TSR. 0: Ignore 1: Set STOP 0 bit to 0	W

12.3.17 Timer Status Register (TSTR)

The TSTR is a 32-bit read-only register. It contains the status of channel in TCU2 mode. The register can be written by setting register TSTSR and TSTCR.

	TSR																BASE+0x0F0															
Bit	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
	Reserved													REAL2	REAL1	Reserved														BUSY2	BUSY1	Reserved
RST	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

Bits	Name	Description	RW
31:19	Reserved	Writing has no effect, read as zero.	R
18	REAL 2	0: The value read from counter 2 is a false value 1: The value read from counter 2 is a real value	R
17	REAL1	0: The value read from counter 1 is a false value 1: The value read from counter 1 is a real value	R
16:3	Reserved	Writing has no effect, read as zero.	R
2	BUSY 2	0: The counter 2 is ready now 1: The counter 2 is busy now	R
1	BUSY1	0: The counter 1 is ready now 1: The counter 1 is busy now	R
0	Reserved	Writing has no effect, read as zero.	R

12.3.18 Timer Status Set Register (TSTSR)

The TSTSR is a 32-bit write-only register. It contains the timer status set bits for each channel.

	TSTSR																BASE+0x0F4															
Bit	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
	Reserved													REALS 2	REALS 1		Reserved													BUSYS 2	BUSYS 1	Reserved
RST	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

Bits	Name	Description	RW
31:19	Reserved	Writing has no effect, read as zero.	R
18	REALS 2	Set REAL 2 bit of TSTR. 0: Ignore 1: Set REAL 2 bit to 1	R
17	REALS 1	Set REAL 1 bit of TSTR. 0: Ignore 1: Set REAL 1 bit to 1	R
16:3	Reserved	Writing has no effect, read as zero.	R
2	BUSYS 2	Set BUSY 2 bit of TSTR. 0: Ignore 1: Set BUSY 2 bit to 1	R
1	BUSYS 1	Set BUSY 1 bit of TSTR. 0: Ignore 1: Set BUSY 1 bit to 1	R
0	Reserved	Writing has no effect, read as zero.	R

12.3.19 Timer Status Clear Register (TSTCR)

The TSTCR is a 32-bit write-only register. It contains the timer status clear bits for each channel.

	TSTCR																BASE+0x0F8															
Bit	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
	Reserved													REALC 2	REALC 1		Reserved													BUSYC 2	BUSYC 1	Reserved
RST	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

Bits	Name	Description	RW
31:19	Reserved	Writing has no effect, read as zero.	R
18	REALC 2	Clear REAL 2 bit of TSTR. 0: Ignore	R

		1: Clear REAL 2 bit to 1	
17	REALC 1	Clear REAL 1 bit of TSTR. 0: Ignore 1: Clear REAL 1 bit to 1	R
16:3	Reserved	Writing has no effect, read as zero.	R
2	BUSYC 2	Clear BUSY 2 bit of TSTR. 0: Ignore 1: Clear BUSY 2 bit to 1	R
1	BUSYC 1	Clear BUSY 1 bit of TSTR. 0: Ignore 1: Clear BUSY 1 bit to 1	R
0	Reserved	Writing has no effect, read as zero.	R

12.4 Operation

12.4.1 Basic Operation in TCU1 Mode

The value of TDFR should be bigger than TDHR, and the minimum settings are TDHR = 0 and TDFR = 1. In this case, the timer output clock cycle is the input clock × 1/2. If TDHR > TDFR, no comparison TFHR signal is generated.

Before the timer counter begin to count up, we need to do as follows:

If you want to use PWM you should set TCSR.PWM_EN to be 0 before you initial TCU.

1 Initial the configuration.
   a Writing TCSR.INITL to initialize PWM output level.
   b Writing TCSR.SD to setting the shutdown mode (Abrupt shutdown or Graceful shutdown).
   c Writing TCSR.PRESCALE to set TCNT count clock frequency.
   d Setting TCNT, TDHR and TDFR.

2 Enable the clock.
   a Writing TCSR.PWM_EN to set whether enable PWM or disable PWM.
   b Writing TCSR.EXT_EN, TCSR.RTC_EN or TCSR.PCK_EN to 1 to select the input clock and enable the input clock. Only one of TCSR.EXT_EN, TCSR.RTC_EN and TCSR.PCK_EN can be set to 1.

After initialize the register of timer, we should start the counter as follows:

3 Enable the counter.
   Setting the TESR.TCST bit to 1 to enable the TCNT.

NOTE: The input clock and PCLK should follow the rules advanced before.

12.4.2 Disable and Shutdown Operation in TCU1 Mode

1 Setting the TECR.TCCL bit to 1 to disable the TCNT.

12.4.3 Basic Operation in TCU2 Mode

The value of TDFR should be bigger than TDHR, and the minimum settings are TDHR = 0 and TDFR = 1. In this case, the timer output clock cycle is the input clock × 1/2. If TDHR > TDFR, no comparison TFHR signal is generated.

Initial state is that TCSR.PRESCALE=0, TCSR.PWM_EN=0 and TCENR=0.

1 Reset the TCU.
   a Writing TCSR.PCK_EN to 1 to select pclk as the input clock.
   b Set TCSR.CLRZ to 1 to clear TCNT or set TCNT to an initial value.
   c Writing TCSR.PCK_EN to 0 to close the input clock.
2 Initial the configuration.
   a Setting TDHR and TDFR.
   b Writing TCSR.INITL to initialize PWM output level (if used PWM).
   c Writing TCSR.PRESCALE to set TCNT count clock frequency.
   d Writing TCSR.EXT_EN, TCSR.RTC_EN or TCSR.PCK_EN to 1 to select the input clock and enable the input clock. Only one of TCSR.EXT_EN, TCSR.RTC_EN and TCSR.PCK_EN can be set to 1.
   e Writing TCSR.PWM_EN to set whether enable PWM or disable PWM.

After initialize the register of timer, we should start the counter as follows:

3 Setting the TESR.TCST bit to 1 to enable the TCNT.

NOTES:

1 You can clear the counter when counter is working.
   a Set TCSR.CLRZ to 1 to clear TCNT.
   b Wait till TSTR.BUSY = 0, that is the counter have been cleared.
2 You can enable PWM or disable PWM the counter when counter is working.
   a Set TCSR.PWM_EN to 1 to enable PWM.
   b Set TCSR.PWM_EN to 0 to disable PWM.

12.4.4 Disable and Shutdown Operation in TCU2 Mode

1 Writing TCSR.PWM_EN to 0 to disable PWM.
2 Setting the TECR.TCCL bit to 1 to disable the TCNT.
3 Wait till TSTR.BUSY = 0, that is the reset of counter is finished.

12.4.5 Read Counter in TCU2 Mode

If you want to read the data from register TCNT when the TCU is working, you can check

TSTR.REAL whether it is good or not. It is suggested that:

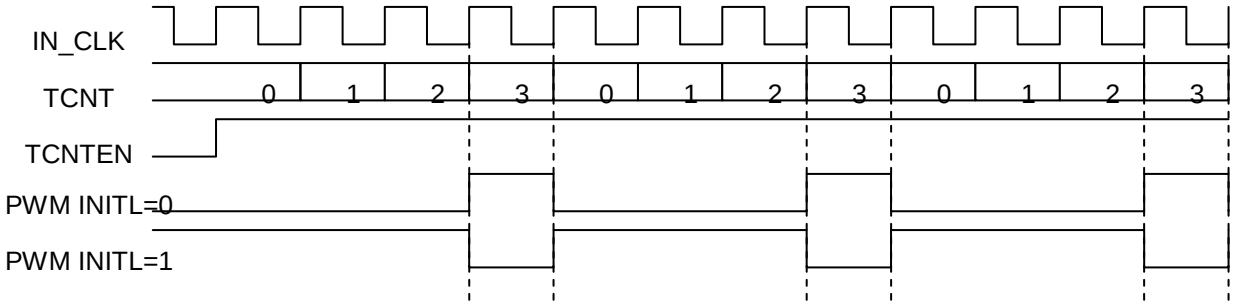
1 If TSTR.REAL==1, the data read is available.
2 If TSTR.REAL==0, reread the counter till TSTR.REAL==1, the data read is available.
3 If TSTR.REAL is always 0, you can read some data, and lose some data that is quick different from the others. Then choose a data from them as the available data.

NOTES:

1 It suggested that (1), (2) is often used when the counter clock is very slow.
2 It suggested that (3) is often used when the counter clock is very fast.

12.4.6 Pulse Width Modulator (PWM)

Timer 0~7 can be used as Pulse Width Modulator (PWM). The PWM can be used to control the back light inverter or adjust bright or contrast of LCD panel.
FULL comparison match signal and HALF comparison match signal can determine an attribute of the PWM_OUT waveform. FULL comparison match signal specifies the clock cycle for the PWM module clock. HALF comparison match signal specifies the duty ratio for the PWM module clock.



12.4.7 Trackball Input Waveform Detect

Timer 0, 3~7 can be used as a waveform edge counter to count both positive edge and negative edge of an external input waveform. For example, a trackball device's input. 4 timers will need to count all four directions (up, down, left, right). You need configure relate GPIO (set relate 4 PWM IO as input) and set relate TCSR.PWM_IN_EN to 1. Both relate TDFR and TDHR need to set to 0xFFFF, unless you need a special interrupt when the counter hit TDFR or TDHR. The counter will clear to 0 when hit TDFR.

Before the timer counter begin to count up, we need to do as follows:

1 Initial the configuration.
  a Writing TCSR.SD to setting the shutdown mode (Abrupt shutdown or Graceful shutdown).
  b Writing TCSR.PRESCALE to set to 0.
  c Setting TCNT, TDHR and TDFR.
2 Enable the clock.
  a Writing TCSR.PWM_EN to disable PWM.

b Writing TCSR.EXT_EN, TCSR.RTC_EN and TCSR.PCK_EN to 0, TCSR.PWM_IN_EN to 1 to select the input clock and enable the input clock.

After initialize the register of timer, we should start the counter as follows:

3 Enable the counter.
Setting the TESR.TCST bit to 1 to enable the TCNT.

NOTE: The input clock and PCLK should follow the rules adviced before.

13 Operating System Timer

13.1 overview

This module include 2 channels, sys_ost32(ost1) and sys_ost64(ost2)

- sys_ost32: 32-bit timer, use to generate operating system's time slice.
  - Has counter Full-match interrupt.
  - Counter by an chip external clock(exclk), which can be divide by 1,4,16
  - Read by an asynchronize system clock.
- sys_ost64: 64-bit timer.
  - Counter by an chip external clock(exclk), which can be divide by 1,4,16
  - Read by an asynchronize system clock

13.2 register description

Following chapter will descript the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute
   R - Read only
   W - Write only
   RW - read and write
   RCW - read and write, but clear to 0 by read
   RSW - read and write, but set to 1 by read
   RWC - read and write, clear to 0 by write 0, write 1 has no effect
   RWS - read and write, set to 1 by write 1, write 0 has no effect
   RC - read only, and clear to 0 by read
   RS - read only, and set to 1 by read
   SPEC - special access method, relate to its description
3. Reset Value
   1 - reset to 1
   0 - reset to 0
   ? - value unknown after reset

Table 13-1 Registers Memory Map-Address Base

Name	Base	Description
SYS_OST	0x12000000	Address base of SYS_OST

Table 13-2 SYS_OST Registers Configuration

Name	Description	RW	Reset Value	Address offset	Access Size
OSTCCR	OS Timer Clock Control Register	RW	0x00000000	0x0000	32
OSTER	OS Timer Counter Enable Register	R	0x00000000	0x0004	32
OSTESR	OS Timer Counter Enable Set Register	W	0x????????	0x0034	32
OSTECR	OS Timer Counter Enable Clear Register	W	0x????????	0x0038	32
OSTCR	OS Timer clear Register	RW	0x00000000	0x0008	32
OSTFR	OS Timer Flag Register	R	0x00000000	0x000C	32
OSTMR	OS Timer Mask Register	RW	0x00000001	0x0010	32
OST1DFR	OS Timer1 Data FULL Register	RW	0xFFFFFFFF	0x0014	32
OST1CNT	OS Timer1 Counter	RW	0x????????	0x0018	32
OST2CNTL	OS Timer2 Counter Lower 32 Bits	RW	0x????????	0x0020	32
OSTCNT2HBUF	OS Timer2 Counter Higher 32 Bits Buffer	R	0x????????	0x0024	32

13.2.1 Timer Clock Control Register (OSTCCR)

	OSTCCR																BASE+0x0000															
Bit	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
	Reserved																												PRESCALE2		PRESCALE1		
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:6	Reserved	Writing has no effect, read as zero.	R
5:3	PRESCALE2	These bits select the TCNT count clock frequency for ost2. Don't change this field when the channel is running. Bit1 / Bit 0 / Description: 0 / 0 / Internal clock: CLK/1 0 / 1 / Internal clock: CLK/4 1 / 0 / Internal clock: CLK/16 1 / 1 / Reserved	RW
2:0	PRESCALE1	These bits select the TCNT count clock frequency for ost1. Don't change this field when the channel is running.	RW

Bit1	Bit 0	Description
0	0	Internal clock: CLK/1
0	1	Internal clock: CLK/4
1	0	Internal clock: CLK/16
1	1	Reserved

13.2.2 Timer Counter Enable Register (OSTER, OSTESR, OSTECR)

The OSTER is a 32-bit read-only register. It contains the counter enable control bits for both channels. It is initialized to 0x00000000 by any reset. It can only be set by register OSTESR and OSTECR. Since the timer enable control bits are located in the same addresses, two timers can be started at the same time. (Default: disable)

	OSTER																BASE+0x0004														
Bit	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
	Reserved																														OST2EN	OST1EN
RST	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

Bits	Name	Description	RW
31:2	Reserved	Writing has no effect, read as zero.	R
1	OST2EN	Enable the counter in channel2's timer. 0: Stop counting up 1: Begin counting up	R
0	OST1EN	Enable the counter in channel1's timer 0: Stop counting up 1: Begin counting up	R

	OSTESR																BASE+0x0034														
Bit	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
	Reserved																														OST2ENS	OST1ENS
RST	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

Bits	Name	Description	RW
31:2	Reserved	Writing has no effect, read as zero.	R
1	OST2ENS	Enable the counter in channel2's timer. 0: no effect	W

		1: set OSTER.OST2EN to 1	
0	OST1ENS	Enable the counter in channel1's timer 0: no effect 1: set OSTER.OST1EN to 1	W

	OSTECR																BASE+0x0038															
Bit	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
	Reserved																														OST2ENC	OST1ENC
RST	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

Bits	Name	Description	RW
31:2	Reserved	Writing has no effect, read as zero.	R
1	OST2ENC	Disable the counter in channel2's timer. 0: no effect 1: clear OSTER.OST2EN to 0	W
0	OST1ENC	Disable the counter in channel1's timer 0: no effect 1: clear OSTER.OST1EN to 0	W

13.2.3 Timer Counter Clear Register (OSTCR)

The OSTCR is a 32-bit read-write register. Write 1 to each bit to clear relate counter.

	OSTCR																BASE+0x0008															
Bit	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
	Reserved																														OST2CLR	OST1CLR
RST	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

Bits	Name	Description	RW
31:2	Reserved	Writing has no effect, read as zero.	R
1	OST2CLR	Clear the counter in channel2's timer. Write 1 at this bit to clear channel2's timer	RW
0	OST1CLR	Clear the counter in channel1's timer. Write 1 at this bit to clear channel1's timer	RW

13.2.4 Timer Data FULL Register (OST1DFR)

The comparison data FULL registers OST1DFR is used to store the data to be compared with the content of the counter OSTCNT.

	OST1DFR																BASE+0x0014															
Bit	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
	OST1DFR																															
RST	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

13.2.5 Timer Counter (OST1CNT)

OST1CNT is a 32-bit read/write register. The up-counter OST1CNT can be reset to 0 by software and counts up using the prescaler output clock. When OST1CNT count up to equal to OSTDFR, it will reset to 0 and continue to count up.

The counter data can be read out at any time. The data can be clear by OSTCR.OST1CLR. (Default: indeterminate)

	OST1CNT																BASE+0x0018															
Bit	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
	OST1CNT																															
RST	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

13.2.6 Timer Flag Register (OST1FR)

The OST1FR is a 32-bit read-only register. It contains the comparison match flag bits for both channels. It is initialized to 0x00000000 by any reset.

	TFR																BASE+0x000C															
Bit	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
	Reserved																															FFLAG
RST	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

Bits	Name	Description	RW
31:1	Reserved	Writing has no effect, read as zero.	R
0	FFLAG	channel1's FULL comparison match flag. (OST1CNT = OSTDFR) 0: Comparison not match 1: Comparison match	RWC

13.2.7 Timer Mask Register (OST1MR)

The OST1MR is a 32-bit read-only register. It contains the comparison match flag bits for both channels. It is initialized to 0x00000001 by any reset.

	OST1MR																BASE+0x0010														
Bit	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
	Reserved																															FMASK
RST	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	Name	Description	RW
31:0	Reserved	Writing has no effect, read as zero.	R
0	FMASK	channel1's FULL comparison match interrupt mask. 0: Comparison match interrupt be not masked 1: Comparison match interrupt be masked	RW

13.2.8 Operating System Timer Counter (OST2CNTH, OST2CNTL)

The operating system timer counter (OST2CNT) is a 64-bit read/write counter. OST2CNTH is the high 32 bits and OST2CNTL is the low 32 bits. The up-counter OSTCNT can be enable by software and counts up using the prescaler output clock. When read OST2CNTL, the OSTCNTH is store at OST2CNTBUF, the software can read OST2CNTBUF to get the total 64-bit number.

	OST2CNTH																BASE+0x001C														
Bit	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
	OST2CNTH																															
RST	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

	OST2CNTL																BASE+0x0020														
Bit	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
	OST2CNTL																															
RST	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

13.2.9 Operating System Timer Counter high 32 bits buffer (OSTCNT2HBUF)

The operating system timer counter high 32 bits buffer OSTCNT2HBUF is used to store the high 32

bits of OSTCNT2 when its lower 32 bits are read by software. This register can be directly read. (Default: indeterminate)

	OST2CNTHBUF																BASE+0x0024															
Bit	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
	OSTCNT2HBUF																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

13.3 Operation

13.3.1 Basic Operation in channel1

1.Initial the configuration.
Writing OSTCCR.PRESCALE to set OST1CNT OST2CNT count clock frequency.
Writing OSTCR.OST1CLR and OST2CLR to set the begin number to 0.

2.After initialize the register of timer, we should start the counter as follows:
Setting the OSTESR.OST1EN(and/or OST2EN) bit to 1 to enable the OST1CNT(and/or OST2CNT). Then it will begin to count up.

13.3.2 Stop Operation in channel1

Set the OSTER.OST1EN 0 to disable the OST1CNT.
Set the OSTER.OST2EN 0 to disable the OST2CNT.

13.3.3 modify the prsecale in channel1

1.It's not permited to modify the prescale when the counter is running.
2.if you want to modify the prescale, you should
first: Setting the OSTER.OST1EN bit to 0 to disable the OST1CNT. Set the OSTER.OST2EN bit to 0 to disable the OSTCNT
second: modify the prescale in OSTCCR
third: Setting the OSTESR.TCST bit to 1 to enable the TCNT again. Setting the TESR.OSTST bit to 1 to enable the OSTCNT again.

14 Interrupt Controller

14.1 Overview

This chapter describes the interrupt controller included in the XBurst Processor, explains its modes of operation, and defines its registers. The interrupt controller controls the interrupt sources available to the processor and contains the location of the interrupt source to allow software to determine source of all interrupts. It also determines whether an IRQ occurs or not and masks the interrupts.

Features:

- Total 64 interrupt sources
- Each interrupt source can be independently enabled
- Priority mechanism to indicate highest priority interrupt
- All the registers are accessed by CPU
- Unmasked interrupts can wake up the chip in sleep mode
- Another set of source, mask and pending registers to serve for PDMA

14.2 Register Description

Following chapter will descript the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute
   R - Read only
   W - Write only
   RW - read and write
   RCW - read and write, but clear to 0 by read
   RSW - read and write, but set to 1 by read
   RWC - read and write, clear to 0 by write 1, write 0 has no effect
   RWS - read and write, set to 1 by write 1, write 0 has no effect
   RC - read only, and clear to 0 by read
   RS - read only, and set to 1 by read
   SPEC - special access method, relate to its description
3. Reset Value
   1 - reset to 1
   0 - reset to 0
   ? - value unknown after reset

Table 14-1 Registers Memory Map-Address Base

Name	Base	Description
INTC	0x10010000	Address base of INTC

Table 14-2 lists the registers of Interrupt Controller. All of these registers are 32bit, and each bit of the register represents or controls one interrupt source that list in Table 14-2.

Table 14-2 INTC Register

Name	Description	RW	Reset Value	Address offset	Access Size
ICSR0	IRQ Source Register	R	0x00000000	0x00	32
ICMR0	IRQ Mask Register	RW	0xFFFFFFFF	0x04	32
ICMSR0	IRQ Mask Set Register	W	0x????????	0x08	32
ICMCR0	IRQ Mask Clear Register	W	0x????????	0x0C	32
ICPR0	IRQ Pending Register	R	0x00000000	0x10	32
ICSR1	IRQ Source Register	R	0x00000000	0x20	32
ICMR1	IRQ Mask Register	RW	0xFFFFFFFF	0x24	32
ICMSR1	IRQ Mask Set Register	W	0x????????	0x28	32
ICMCR1	IRQ Mask Clear Register	W	0x????????	0x2C	32
ICPR1	IRQ Pending Register	R	0x00000000	0x30	32
DSR0	IRQ Source Register0 for PDMA	R	0x00000000	0x34	32
DMR0	IRQ Mask Register0 for PDMA	RW	0xFFFFFFFF	0x38	32
DPR0	IRQ Pending Register0 for PDMA	R	0x00000000	0x3C	32
DSR1	IRQ Source Register1 for PDMA	R	0x00000000	0x40	32
DMR1	IRQ Mask Register1 for PDMA	RW	0xFFFFFFFF	0x44	32
DPR1	IRQ Pending Register1 for PDMA	R	0x00000000	0x48	32

14.2.1 Interrupt Controller Source Register (ICSR0)

This register contains all the interrupts' status. A "1" indicates that the corresponding interrupt is pending . A "0" indicates that the interrupt is not pending now. The register is read only.

	ICSR0																														0x10001000
Bit	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
	LCD	CIM	Reserved	Reserved	TCU0	TCU1	TCU2	Reserved	AES	Reserved	OTG	Reserved	Reserved	Reserved	GPIO0	GPIO1	GPIO2	GPIO3	Reserved	Reserved	PDMAD	PDMA	Reserved	SSI0	SFC	Reserved	Reserved	Reserved	Reserved	Reserved	AIC0	DMIC
RST	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

Bits Of ICSR0	Description
0	The corresponding interrupt source is not pending.
1	The corresponding interrupt source is pending.

14.2.2 Interrupt Controller Source Register (ICSR1)

This register contains all the interrupts' status. A "1" indicates that the corresponding interrupt is pending . A "0" indicates that the interrupt is not pending now. The register is read only.

	ICSR1																													0x10001020
Bit	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
	Reserved	JPEG	PDMAM	I2C0	I2C1	I2C2	Reserved	Reserved	MAC	EFUSE	Reserved	DDR	UART0	UART1	UART2	Reserved	CPM	HARB0	Reserved	HARB2	Reserved			PCM0	Reserved	SCC	MSC0	MSC1	Reserved	Reserved	Reserved	RTC
RST	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

Bits Of ICSR1	Description
0	The corresponding interrupt source is not pending.
1	The corresponding interrupt source is pending.

14.2.3 Interrupt Controller Mask Register (ICMR0)

This register is used to mask the interrupt input sources and defines which active sources are allowed to generate interrupt requests to the processor. Its value can be changed either by writing ICMSR and ICMCR or by writing itself. The masked interrupts are invisible to the processor.

	ICMR0																													0x10001004
Bit	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
	LCD	CIM	Reserved	Reserved	TCU0	TCU1	TCU2	Reserved	AES	Reserved	OTG	Reserved	Reserved	Reserved	GPIO0	GPIO1	GPIO2	GPIO3	Reserved	Reserved	PDMAD	PDMA	Reserved	SSI0	SFC	Reserved	Reserved	Reserved	Reserved	Reserved	AIC0	DMIC
RST	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits Of ICMR0	Description
0	The corresponding interrupt is not masked.
1	The corresponding interrupt is masked.

14.2.4 Interrupt Controller Mask Register (ICMR1)

This register is used to mask the interrupt input sources and defines which active sources are allowed to generate interrupt requests to the processor. Its value can be changed either by writing ICMSR and ICMCR or by writing itself. The masked interrupts are invisible to the processor.

	ICMR1																													0x10001024
Bit	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
	Reserved	JPEG	PDMAM	I2C0	I2C1	I2C2	Reserved	Reserved	MAC	EFUSE	Reserved	DDR	UART0	UART1	UART2	Reserved	CPM	HARB0	Reserved	HARB2	Reserved			PCM0	Reserved	SCC	MSC0	MSC1	Reserved	Reserved	Reserved	RTC
RST	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

Bits Of ICMR1	Description
0	The corresponding interrupt is not masked.
1	The corresponding interrupt is masked.

14.2.5 Interrupt Controller Mask Set Register (ICMSR0)

This register is used to set bits in the interrupt mask register. This register is write only.

	ICMSR0																													0x10001008
Bit	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
	LCD	CIM	Reserved	Reserved	TCU0	TCU1	TCU2	Reserved	AES	Reserved	OTG	Reserved	Reserved	Reserved	GPIO0	GPIO1	GPIO2	GPIO3	Reserved	Reserved	PDMAD	PDMA	Reserved	SSI0	SFC	Reserved	Reserved	Reserved	Reserved	Reserved	AIC0	DMIC
RST	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

Bits Of ICMSR0	Description
0	Ignore.
1	Will set the corresponding interrupt mask bit.

14.2.6 Interrupt Controller Mask Set Register (ICMSR1)

This register is used to set bits in the interrupt mask register. This register is write only.

	ICMSR1																													0x10001028
Bit	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
	Reserved	JPEG	PDMAM	I2C0	I2C1	I2C2	Reserved	Reserved	MAC	EFUSE	Reserved	DDR	UART0	UART1	UART2	Reserved	CPM	HARB0	Reserved	HARB2	Reserved			PCM0	Reserved	SCC	MSC0	MSC1	Reserved	Reserved	Reserved	RTC
RST	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

Bits Of ICMSR1	Description
0	Ignore.
1	Will set the corresponding interrupt mask bit.

14.2.7 Interrupt Controller Mask Clear Register (ICMCR0)

This register is used to clear bits in the interrupt mask register. This register is write only.

	ICMCR0																														0x1000100C
Bit	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
	LCD	CIM	Reserved	Reserved	TCU0	TCU1	TCU2	Reserved	AES	Reserved	OTG	Reserved	Reserved	Reserved	GPIO0	GPIO1	GPIO2	GPIO3	Reserved	Reserved	PDMAD	PDMA	Reserved	SSI0	SFC	Reserved	Reserved	Reserved	Reserved	Reserved	AIC0	DMIC
RST	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

Bits Of ICMCR0	Description
0	Ignore.
1	Will clear the corresponding interrupt mask bit.

14.2.8 Interrupt Controller Mask Clear Register (ICMCR1)

This register is used to clear bits in the interrupt mask register. This register is write only.

	ICMCR1																														0x1000102C
Bit	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
	Reserved	JPEG	PDMAM	I2C0	I2C1	I2C2	Reserved	Reserved	MAC	EFUSE	Reserved	DDR	UART0	UART1	UART2	Reserved	CPM	HARB0	Reserved	HARB2	Reserved			PCM0	Reserved	SCC	MSC0	MSC1	Reserved	Reserved	Reserved	RTC
RST	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

Bits Of ICMCR1	Description
0	Ignore.
1	Will clear the corresponding interrupt mask bit.

14.2.9 Interrupt Controller Pending Register (ICPR0)

This register contains the status of the interrupt sources after masking. This register is read only.

	ICPR0																														0x10001010
Bit	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
	LCD	CIM	Reserved	Reserved	TCU0	TCU1	TCU2	Reserved	AES	Reserved	OTG	Reserved	Reserved	Reserved	GPIO0	GPIO1	GPIO2	GPIO3	Reserved	Reserved	PDMAD	PDMA	Reserved	SSI0	SFC	Reserved	Reserved	Reserved	Reserved	Reserved	AIC0	DMIC
RST	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

Bits Of ICPR0	Description
0	The corresponding interrupt is not active or is masked.
1	The corresponding interrupt is active and is not masked to the processor.

14.2.10 Interrupt Controller Pending Register (ICPR1)

This register contains the status of the interrupt sources after masking. This register is read only.

	ICPR1																														0x10001030
Bit	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
	Reserved	JPEG	PDMAM	I2C0	I2C1	I2C2	Reserved	Reserved	MAC	EFUSE	Reserved	DDR	UART0	UART1	UART2	Reserved	CPM	HARB0	Reserved	HARB2	Reserved			PCM0	Reserved	SCC	MSC0	MSC1	Reserved	Reserved	Reserved	RTC
RST	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

Bits Of ICPR1	Description
0	The corresponding interrupt is not active or is masked.
1	The corresponding interrupt is active and is not masked to the processor.

14.2.11 Interrupt Source Register0 for PDMA (DSR0)

This register contains status of all interrupts. A “1” indicates the corresponding interrupt is pending. The register is read only.

	DSR0																														0x10001034
Bit	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
	LCD	CIM	Reserved	Reserved	TCU0	TCU1	TCU2	Reserved	AES	Reserved	OTG	Reserved	Reserved	Reserved	GPIO0	GPIO1	GPIO2	GPIO3	Reserved	Reserved	PDMAD	PDMA	Reserved	SSI0	SFC	Reserved	Reserved	Reserved	Reserved	Reserved	AIC0	DMIC
RST	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

Bits Of DSR0	Description
0	The corresponding interrupt source is not pending.
1	The corresponding interrupt source is pending.

14.2.12 Interrupt Mask Register0 for PDMA (DMR0)

This register is used to mask the interrupt input sources and defines which active sources are allowed to generate interrupt requests to the processor. The masked interrupts are invisible to the processor.

	DMR0																0x10001038													
Bit	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
	LCD	CIM	Reserved	Reserved	TCU0	TCU1	TCU2	Reserved	AES	Reserved	OTG	Reserved	Reserved	Reserved	GPIO0	GPIO1	GPIO2	GPIO3	Reserved	Reserved	PDMAD	PDMA	Reserved	SSI0	SFC	Reserved	Reserved	Reserved	Reserved	Reserved	AIC0	DMIC
RST	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

Bits Of DMR0	Description
0	The corresponding interrupt source is not pending.
1	The corresponding interrupt source is pending.

14.2.13 Interrupt Pending Register0 for PDMA (DPR0)

This register contains the status of the interrupt sources after masking. This register is read only.

	DPR0																															
																															0x1000103C	
Bit	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
	LCD	CIM	Reserved	Reserved	TCU0	TCU1	TCU2	Reserved	AES	Reserved	OTG	Reserved	Reserved	Reserved	GPIO0	GPIO1	GPIO2	GPIO3	GPIO4	GPIO5	PDMAD	PDMA	Reserved	SSI0	SFC	Reserved	Reserved	Reserved	Reserved	Reserved	AIC0	DMIC
RST	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

Bits Of ICPR0	Description
0	The corresponding interrupt is not active or is masked.
1	The corresponding interrupt is active and is not masked to the processor.

14.2.14 Interrupt Source Register1 to PDMA (DSR1)

This register contains status of all interrupts. A “1” indicates the corresponding interrupt is pending.

The register is read only.

	DSR1																														0x10001040
Bit	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
	Reserved	JPEG	PDMAM	I2C0	I2C1	I2C2	Reserved	Reserved	MAC	EFUSE	Reserved	DDR	UART0	UART1	UART2	Reserved	CPM	HARB0	Reserved	HARB2	Reserved			PCM0	Reserved	SCC	MSC0	MSC1	Reserved	Reserved	Reserved	RTC
RST	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

Bits Of DSR1	Description
0	The corresponding interrupt source is not pending.
1	The corresponding interrupt source is pending.

14.2.15 Interrupt Mask Register1 for PDMA (DMR1)

This register is used to mask the interrupt input sources and defines which active sources are allowed to generate interrupt requests to the processor. The masked interrupts are invisible to the processor.

	DMR1																														0x10001044
Bit	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
	Reserved	JPEG	PDMAM	I2C0	I2C1	I2C2	Reserved	Reserved	MAC	EFUSE	Reserved	DDR	UART0	UART1	UART2	Reserved	CPM	HARB0	Reserved	HARB2	Reserved			PCM0	Reserved	SCC	MSC0	MSC1	Reserved	Reserved	Reserved	RTC
RST	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

Bits Of DMR1	Description
0	The corresponding interrupt is not masked.
1	The corresponding interrupt is masked.

14.2.16 Interrupt Pending Register1 for PDMA (DPR1)

This register contains the status of the interrupt sources after masking. This register is read only.

	DPR1																														0x10001048
Bit	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
	Reserved	JPEG	PDMAM	I2C0	I2C1	I2C2	Reserved	Reserved	MAC	EFUSE	Reserved	DDR	UART0	UART1	UART2	Reserved	CPM	HARB0	Reserved	HARB2	Reserved			PCM0	Reserved	SCC	MSC0	MSC1	Reserved	Reserved	Reserved	RTC
RST	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

Bits Of ICPR0	Description
0	The corresponding interrupt is not active or is masked.
1	The corresponding interrupt is active and is not masked to the processor.

14.3 Software Considerations

The interrupt controller is reflecting the status of interrupts sources in the peripheral.

Software should perform the task - determine the interrupt source from in ICPRx. In this chip, pending interrupts have two levels in structure. Interrupting module in the system that contains more than one interrupt sources need software to determine how to service it by reading interrupt status registers within it.

In the interrupt handler, the serviced interrupt source needs to be cleared in the interrupting device. In order to make certain the cleared source request status has been reflected at the corresponding ICPRx bit, software should wait enough time before exiting interrupt state.

The procedure is described following:

1 Interrupt generated.
2 CPU query interrupt sources, saves the current environment and then goes to interrupt common service routine.
3 Get ICPRx.
4 Find the highest priority interrupt and vector it. (The software decides which one has the highest priority)
5 Mask the chosen interrupt by writing the register ICMSRx.
6 Enable the system interrupt to allow the interrupt nesting. (software decided)
7 Execute the interrupt handler and unmask it by writing the register ICMCRx when exit the handler.
8 CPU restores the saved environment and exits the interrupt state.

15 Watchdog Timer

15.1 Overview

The watchdog timer is used to resume the processor whenever it is disturbed by malfunctions such as noise and system errors. The watchdog timer can generate the reset signal.

Features:

- Generates WDT reset
- A 16-bit Data register and a 16-bit counter
- Counter clock uses the input clock selected by software
  - PCLK, EXTAL and RTCCLK can be used as the clock for counter
  - The division ratio of the clock can be set to 1, 4, 16, 64, 256 and 1024 by software

15.2 Register Description

Following chapter will descript the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute
   R - Read only
   W - Write only
   RW - read and write
   RCW - read and write, but clear to 0 by read
   RSW - read and write, but set to 1 by read
   RWC - read and write, clear to 0 by write 1, write 0 has no effect
   RWS - read and write, set to 1 by write 1, write 0 has no effect
   RC - read only, and clear to 0 by read
   RS - read only, and set to 1 by read
   SPEC - special access method, relate to its description
3. Reset Value
   1 - reset to 1
   0 - reset to 0
   ? - value unknown after reset

Table 15-1 Registers Memory Map-Address Base

Name	Base	Description
WDT	0x10002000	Address base of WDT

Table 15-2 WDT Registers Configuration

Name	Description	RW	Reset Value	Address offset	Access Size
TDR	Watchdog Timer Data Register	RW	0x????	0x000	16
TCER	Watchdog Counter Enable Register	RW	0x00	0x004	8
TCNT	Watchdog Timer Counter	RW	0x????	0x008	16
TCSR	Watchdog Timer Control Register	RW	0x0000	0x00C	16

15.2.1 Watchdog Control Register (TCSR)

The TCSR is a 16-bit read/write register. It contains the control bits for WDT. It is initialized to 0x00 by any reset.

TCSR																BASE+0x 00C															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																Reserved										PRESCALE			EXT_EN	RTC_EN	PCK_EN
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:6	Reserved	Writing has no effect, read as zero.	R
5:3	PRESCALE	These bits select the TCNT count clock frequency.	RW

Bit 2	Bit1	Bit 0	Description
0	0	0	Internal clock: CLK/1
0	0	1	Internal clock: CLK/4
0	1	0	Internal clock: CLK/16
0	1	1	Internal clock: CLK/64
1	0	0	Internal clock: CLK/256
1	0	1	Internal clock: CLK/1024
110~111			Reserved

2	EXT_EN	Select EXTAL as the timer clock input. 1: Enable 0: Disable	RW
1	RTC_EN	Select RTCCLK as the timer clock input. 1: Enable 0: Disable	RW
0	PCK_EN	Select PCLK as the timer clock input. 1: Enable 0: Disable	RW

NOTE: The input clock of timer and the PCLK should keep to the rules as follows:

Input clock of timer: IN_CLK	Clock generated from the frequency divider (PRESCALE): DIV_CLK
PCK_EN == 0, RTC_EN == 1 and EXT_EN == 0 (IN_CLK = RTCCLK)	$f_{DIV\_CLK} < ½ f_{PCLK}$
PCK_EN == 0, RTC_EN == 0 and EXT_EN == 1 (IN_CLK = EXTAL)	$f_{DIV\_CLK} < ½ f_{PCLK}$
PCK_EN == 1, RTC_EN == 0 and EXT_EN == 0 (IN_CLK = PCLK)	ANY

15.2.2 Watchdog Enable Register (TCER)

The TCER is an 8-bit read/write register. It contains the counter enable control bits for watchdog. It is initialized to 0x00 by any reset.

	TCER																BASE+0x 004							
Bit																	7	6	5	4	3	2	1	0
	Reserved																							TCEN
RST																	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:1	Reserved	Writing has no effect, read as zero.	R
0	TCEN	Counter enable control. 0: Timer stop 1: Timer running	RW

Note: writing TCEN to be zero can not stop the counter, you must stop it by writing TCU's register TSSR.WDTSS high.That is to say the counter can run when TCEN high and TCU's TSR low, but it can only stop by set TCU's TSR high.

15.2.3 Watchdog Timer Data Register (TDR)

The watchdog timer data register TDR is used to store the data to be compared with the content of the watchdog timer up-counter TCNT. This register can be directly read and written. (Default: indeterminate)

	TDR																BASE+0x 000							
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	TDR															
RST																	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

15.2.4 Watchdog Timer Counter (TCNT)

The watchdog timer counter (TCNT) is a 16-bit read/write counter. The up-counter TCNT can be reset to 0 by software and counts up using the prescaler output clock. When TCNT count up to equal to TDR, the comparison match signal will be generated and a WDT reset is generated. The data can be read out at any time. The counter data can be written at any time. (Default: indeterminate)

	TCNT																BASE+0x 008															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	TCNT															
RST																	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

15.3 Watchdog Timer Function

The following describes steps of using WDT:

1 Setting the PRESCALE of input clock in register TCSR.
2 Set register TDR and TCNT.
3 Select the input clock and enable the input clock in register TCSR.

After initialize the register of timer, we should start the counter as follows:

4 Set TCEN bit in TCER to 1. The counter TCNT begins to count.
5 If TCNT = TDR, a WDT reset will be generated.

NOTES:

1 The input clock and PCLK should follow the rules advanced before.
2 The clock of WDT can be stopped by setting register TSR, and register TSR can only be set by register TSSR or TSCR. The content of register TSR, TSSR and TSCR can be found in TCU spec.

16 PDMA Controller

16.1 Overview

Programmable DMA controller (PDMAC) is dedicated to smartly transfer data between on-chip peripherals (MSC, AIC, UART, etc.), external memories, and memory-mapped external devices.

16.2 Features

- Support up to 8 independent DMA channels
- Descriptor or No-Descriptor Transfer mode compatible with previous Ingenic SOC
- A simple Xburst-1 CPU supports smart transfer mode controlled by programmable firmware
- Transfer data units: 1-byte, 2-byte, 4-byte, 16-byte, 32-byte, 64-byte, 128-byte
- Transfer number of data unit: 1 ~ 2^{24} - 1
- Independent source and destination port width: 8-bit, 16-bit, 32-bit
- Fixed three priorities of channel groups: 0~3, highest; 4~11: mid; 12~31: lowest
- An extra INTC IRQ can be bound to one programmable DMA channel

16.3 Block Diagram

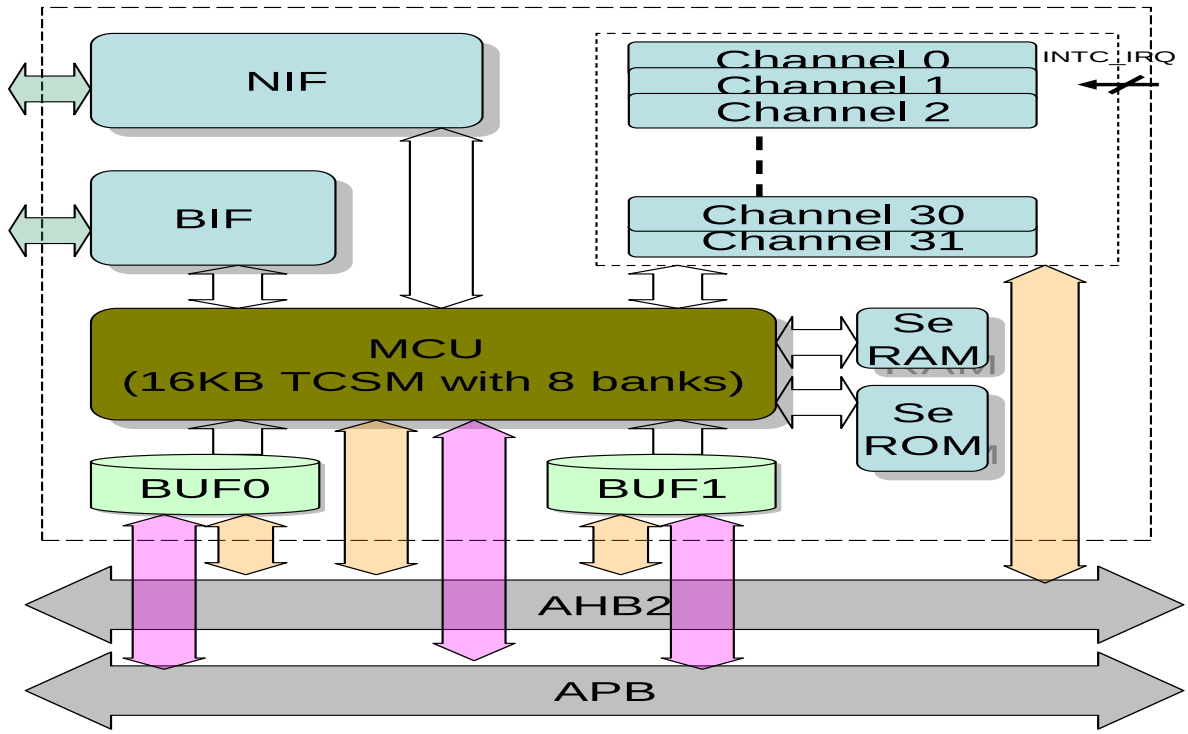


Figure 16-1 Block Diagram of PDMA

16.4 Memory Mapped Register Descriptions

16.4.1 DMA Channel Registers

Following chapter will descript the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute
   R - Read only
   W - Write only
   RW - read and write
   RCW - read and write, but clear to 0 by read
   RSW - read and write, but set to 1 by read
   RWC - read and write, clear to 0 by write 1, write 0 has no effect
   RWS - read and write, set to 1 by write 1, write 0 has no effect
   RC - read only, and clears to 0 by read
   RS - read only, and set to 1 by read
   SPEC - special access method, relate to its description
3. Reset Value
   1 - Reset to 1
   0 - reset to 0
   ? - Value unknown after reset

Table 16-1 Registers Memory Map-Address Base

Name	Base	Description
Channel n	0x13420000 + n*0x20	PDMA channel n.

Table 16-2 DMA Channel Registers (n=0~31)

Name	Description	Reset Value	Address offset	Access Size (bit)
DSAn	Channel n Source Address	0x?	0x00 + n*0x20	32
DTAn	Channel n Target Address	0x?	0x04 + n*0x20	32
DTCn	Channel n Transfer Count	0x?	0x08 + n*0x20	32
DRTn	Channel n Request Source	0x?	0x0C + n*0x20	32
DCSn	Channel n Control/Status	0x?	0x10 + n*0x20	32
DCMn	Channel n Command	0x?	0x14 + n*0x20	32
DDAn	Channel n Descriptor Address	0x?	0x18 + n*0x20	32
DSDn	Channel n Stride Difference	0x?	0x1C + n*0x20	32

16.4.2 Global Control Registers

Table 16-3 Registers Memory Map-Address Base

Name	Base	Description
GCR	0x13420000	PDMA global control registers.

Table 16-4 Global Control Registers

Name	Description	Reset Value	Address offset	Access Size (bit)
DMAC	DMA Control	0x0	0x1000	32
DIRQP	DMA Interrupt Pending	0x0	0x1004	32
DDB	DMA Doorbell	0x0	0x1008	32
DDS	DMA Doorbell Set	0x0	0x100C	32
DIP	Descriptor Interrupt Pending	0x0	0x1010	32
DIC	Descriptor Interrupt Clear	0x0	0x1014	32
DMACP	DMA Channel Programmable	0x0	0x101C	32
DSIRQP	Channel soft IRQ to MCU	0x0	0x1020	32
DSIRQM	Channel soft IRQ mask	0xffffffff	0x1024	32
DCIRQP	Channel IRQ to MCU	0x0	0x1028	32
DCIRQM	Channel IRQ to MCU mask	0xffffffff	0x102C	32
DMCS	MCU Control and Status	0xe1	0x1030	32
DMNMB	MCU Normal Mailbox	0x0	0x1034	32
DMSMB	MCU Security Mailbox	0x0	0x1038	32
DMINT	MCU Interrupt	0x3	0x103C	32

NOTES:

Grey ones are obsolete registers defined in previous Ingenic SOC. They are relative to clock gating and have no real function, and they are not supported any longer.

16.5 DMA Channel Register Definition

16.5.1 DMA Source Address (DSAn, n = 0 ~ 7)

	DSAn																BASE + 0x0 + (n*0x20)													
Bit	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
	SA																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:0	SA	Source physical address.	RW

16.5.2 DMA Target Address (DTAn, n = 0 ~ 7)

	DTAn																BASE + 0x4 + (n*0x20)													
Bit	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
	TA																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:0	TA	Target physical address.	RW

16.5.3 DMA Transfer Count (DTCn, n = 0 ~ 7)

	DTCn																BASE + 0x8 + (n*0x20)													
Bit	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
	Reserved								TC																							
RST	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:24	Reserved	Write has no effect, read as zero.	R
23:0	TC	TC records the number of data unit to be transferred. Moreover, when Stride transfer mode is enabled, TC composes of two parts: the lower 16 bits is the number of data unit for sub-block transfer, the higher 8 bits is the number of sub-block. TC automatically counts down to 0 at the end.	RW

16.5.4 DMA Request Types (DRTn, n = 0 ~ 7)

	DRTn																BASE + 0xC + (n*0x20)													
Bit	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
	Reserved																										RT					
RST	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	?	?	?	?	?	?

Bits	Name	Description	RW
31:6	Reserved	Write has no effect, read as zero.	R
5:0	RT	Transfer request type.	RW

Table 16-5 Transfer Request Types

RT5-0	Description
000000	Reserved.
000001	Reserved.
000101	DMIC receive-fifo-full transfer request.
000110	I2S0 transmit-fifo-empty transfer request.
000111	I2S0 receive-fifo-full transfer request.
001000	Auto-request. (external address → external address)
010000	UART2 transmit-fifo-empty transfer request. (external address → UTHR)
010001	UART2 receive-fifo-full transfer request. (URBR → external address)
010010	UART1 transmit-fifo-empty transfer request. (external address → UTHR)
010011	UART1 receive-fifo-full transfer request. (URBR → external address)
010100	UART0 transmit-fifo-empty transfer request. (external address → UTHR)
010101	UART0 receive-fifo-full transfer request. (URBR → external address)
010110	SSI0 transmit-fifo-empty transfer request.
010111	SSI0 receive-fifo-full transfer request.
011010	MSC0 transmit-fifo-empty transfer request.
011011	MSC0 receive-fifo-full transfer request.
011100	MSC1 transmit-fifo-empty transfer request.
011101	MSC1 receive-fifo-full transfer request.
100000	PCM0 transmit-fifo-empty transfer request.
100001	PCM0receive-fifo-full transfer request.
100100	SMB0 transmit-fifo-empty transfer request.
100101	SMB0 receive-fifo-full transfer request.
100110	SMB1 transmit-fifo-empty transfer request.
100111	SMB1 receive-fifo-full transfer request.
101000	SMB2 transmit-fifo-empty transfer request.
101001	SMB2 receive-fifo-full transfer request.

NOTES:

1 Only auto request can be concurrently set in all channels with different source and target address.

2 For on-chip device DMA request, the corresponding source or target address that map to on-chip device must be set as fixed.

16.5.5 DMA Channel Control/Status (DCSn, n = 0 ~ 7)

	DCSn																BASE + 0x10 + (n*0x20)													
Bit	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

	NDES	DES8	Reserved	CDOA	Reserved	AR	TT	HLT	Reserved	CTE
RST	?	?	? ? ? ? ? ? ? ? ? ? ? ? ? ?	? ? ? ? ? ? ? ?	0 0 0	0	0	0	0	0

Bits	Name	Description	RW
31	NDES	Descriptor or No-Descriptor Transfer Select. 0: Descriptor Transfer; 1: No-descriptor Transfer.	RW
30	DES8	Descriptor 8 Word. 0: 4-word descriptor; 1: 8-word descriptor.	RW
29:16	Reserved	Write has no effect, read as zero.	R
15:8	CDOA	Copy of offset address of last completed descriptor from that in DMA command register. (The field is Ignored in No-Descriptor Transfer)	RW
7:5	Reserved	Write has no effect, read as zero.	R
4	AR	Address Error. Hardware set it to 1 and software clear it to 0. 0: no address error; 1: address error occurred.	RW
3	TT	Transfer Terminate. 0: Descriptor or No-Descriptor DMA transfer does not end 1: No-Link Descriptor or No-Descriptor DMA transfer end	RW
2	HLT	DMA halt. Hardware sets it to 1 when an abnormal case occurs during transfer, then software has to clear it to 0 for re-using the channel later.	RW
1	Reserved	Write has no effect, read as zero.	R
0	CTE	Channel transfer enable: 0: disable; 1: enable.	RW

NOTES:

1. For a working channel, when an address error occurs or an on-going transfer is stopped by software writing 0 to CTE, HLT becomes 1 but TT keeps 0 that denotes an abnormal situation occurs.
2. Software can stop any working channel at any time. However, after setting 0 to CTE, software must read TT, HLT and CTE to ensure the stop behavior has been done for the channel, that is,
   a) TT=0, HALT=0, CTE =1, not halt yet, need polling again
   b) TT=1, HALT=0, CTE=0, the channel just ends its work when software wants to stop it.
   c) TT=0, HALT=1, CTE=0, the channel has been stopped but its transfer not complete yet.

16.5.6 DMA Channel Command (DCMn, n = 0 ~ 7)

	DCMn																BASE + 0x14 + (n*0x20)													
Bit	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
	Reserved								SAI	DAI	Reserved		RDIL				SP		DP		Reserved		TSZ		Reserved					STDE	TIE	LINK
RST	0	0	0	0	0	0	0	0	?	?	0	0	?	?	?	?	?	?	?	?	0	?	?	?	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:24	Reserved	Write has no effect, read as zero	R
23	SAI	Source Address Increment. 0: no increment; 1: increment.	RW
22	DAI	Destination Address Increment. 0: no increment; 1: increment.	RW
19:16	RDIL	Recommended data unit size (unit: byte) for triggering device's DMA request when TSZ is autonomy.	RW
15:14	SP	Source port width. 00: 32-bit; 01: 8-bit; 10: 16-bit; 11: reserved.	RW
13:12	DP	Destination port width. 00: 32-bit; 01: 8-bit; 10: 16-bit; 11: reserved.	RW
11	Reserved	Write has no effect, read as zero.	R
10:8	TSZ	Transfer Data Size of a data unit. 000: 32-bit; 001: 8-bit; 010: 16-bit; 011: 16-byte; 100: 32-byte; 101: 64-byte; 110: 128-byte; 111: autonomy;	RW
7:3	Reserved	Write has no effect, read as zero.	R
2	STDE	Stride Disable/Enable. 0: address stride disable; 1: address stride enable.	RW
1	TIE	Transfer Interrupt Enable (TIE). 0: disable interrupt; 1: enable interrupt when TT is set to 1.	RW
0	LINK	Descriptor Link Enable. 0: disable; 1: enable. (the field is ignored in No-Descriptor Transfer mode)	RW

NOTES:

SP and DP are only available for device FIFO or NEMC. But when transfer data between device and DDR memory, must set the port width of DDR side as the same as the device side; however, when both source and destination are in DDR memory, both SP and DP must be 32-bit by setting 0.

Table 16-6 Available RDIL

RDIL	Description
0	Reserved
1	Recommended data unit is 1 bytes
2	Recommended data unit is 2 bytes
3	Recommended data unit is 3 bytes
4	Recommended data unit is 4 bytes
5	Recommended data unit is 8 bytes
6	Recommended data unit is 16 bytes
7	Recommended data unit is 32 bytes
8	Recommended data unit is 64 bytes
9	Recommended data unit is 128 bytes
10~15	Reserved

NOTES:

1. if TSZ is autonomy, total bytes to be transferred equal the value of DCTn.TC (so maximum transfer bytes for autonomy are 16MB -1).
2. Programmer must care the detail of the FIFO of the device binding with relative DMA channel to set the correct and best recommended data unit value according to FIFO's data width and depth, refer to following table.

FIFO form	Best recommended RDIL
8x8bits	4 (half of total entries just accommodate 4 bytes)
8x16bits	5 (half of total entries just accommodate 8 bytes)
32x8bits	6 (half of total entries just accommodate 16 bytes)
16x16bits	6 (half of total entries just accommodate 16 bytes)
64x8bits	7 (half of total entries just accommodate 32 bytes)
64x32bits	9 (half of total entries just accommodate 128 bytes)
128x16bits	9 (half of total entries just accommodate 128 bytes)
128x32bits	9 (because maximum data unit is 128 bytes)

Moreover, programmer must be carefully obey following rules for correctness and best performance concern of DMA transfer:

1. ***auto-request, both source and destination device are DDR memory***. No alignment constraint for address, however, both source port width and destination port width must be set 32bit, and transferred bytes in a bus transaction denoted by TSZ (when TSZ !=7) must not be less than 4. Setting autonomy (TSZ=7) is the best choice for transfer efficiency[*0]
2. ***external device fifo receive request, destination device is DDR memory***, then no alignment constraint for destination address, and
   a) if non-autonomy (TSZ !=7), for transferred bytes in a bus transaction denoted by TSZ, it must not exceed the critical value of triggering DMA request[*1] meanwhile must be times of 2-byte when source port width is 16bit or times of 4-byte when source port width is 32bit.
   b) If autonomy (TSZ=7), for transferred bytes in a bus transaction denoted by RDIL, it must not exceed the critical value of triggering DMA request[*1] meanwhile must be times of 2-byte when source port width is 16bit or times of 4-byte when source port width is 32bit.
3. ***external device fifo transmit request, source device is DDR memory***, then no alignment constraint for source address, and
   a) if destination port width is 8bit, setting autonomy (TSZ=7) is the best choice for transfer efficiency
   b) if destination port width is 16bit, when setting autonomy (TSZ=7), total transferred bytes must be the times of 2-byte, and for transferred bytes in a bus transaction denoted by RDIL, it must not exceed the critical value of triggering DMA request[*1] meanwhile must be the times of 2-byte; when setting non-autonomy (TSZ!=7), for transferred bytes in a bus transaction denoted by TSZ, it must not exceed the critical value of triggering DMA request[*1] meanwhile must be the times of 2-byte.
   c) if destination port width is 32bit, when setting autonomy (TSZ=7), total transferred bytes must be the times of 4-byte, and for transferred bytes in a bus transaction denoted by RDIL, it must not exceed the critical value of triggering DMA request[*1] meanwhile must be the times of 4-byte; when setting non-autonomy (TSZ!=7), for transferred bytes in a bus transaction denoted by TSZ, it must not exceed the critical value of triggering DMA request[*1] meanwhile must be the times of 4-byte.

NOTES:

*0 - when TSZ is not autonomy, total bytes to be transferred are bytes represented by TSZ * TC, and each bus transaction can only transfer bytes represented by TSZ, so larger data unit is better for transfer efficiency. But since in practical, in most cases, total bytes to be transferred are not the times of the best data unit, or source and/or destination address violates alignment, thus **setting autonomy for TSZ is recommended** when condition is ok.

*1 - Please refer to relative device's spec for detail information of how to set critical trigger value.

16.5.7 DMA Descriptor Address (DDAn, n = 0 ~ 7)

This register is ignored in No-Descriptor Transfer mode.

	DDAn																BASE + 0x18 + (n*0x20)											
Bit	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
	DBA																DOA												Reserved			
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0

Bits	Name	Description	RW
31:12	DBA	Descriptor Base Address.	RW
11:4	DOA	Descriptor Offset Address.	RW
3:0	Reserved	Write has no effect, read as zero.	R

16.5.8 DMA Stride Difference (DSDn, n = 0 ~ 7)

This register is ignored in No-Descriptor Transfer mode.

When address stride transfer mode is enabled in Descriptor mode, after a sub-block's transfer defined in DTCn completes, the source or target stride difference will be added to perform the new sub-block's start address, and the transfer will keep going until the transfer ends which means TC in DTCn reaches 0.

	DSDn																BASE + 0x1C + (n*0x20)															
Bit	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
	TSD																SSD															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:16	TSD	Target Stride Difference (next sub-block's start address – current sub-block's end address - 1), value range is -32768 ~ 32767	RW
15:0	SSD	Source Stride Difference (next sub-block's start address – current sub-block's end address - 1), value range is -32768 ~ 32767	RW

16.6 DMA Global Register Definition

16.6.1 DMA Control

	DMAC																BASE+0x1000													
Bit	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
	FMSC	FSSI	FTSSI	FUART	FAIC	Reserved																							HLT	AR	Reserved	DMAE
RST	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

Bits	Name	Description	RW
31	FMSC	0: 0.5-data/clk for MSC on APB;1: 1-data/clk for MSC on APB.	RW
30	FSSI	0: 0.5-data/clk for SSI on APB;1: 1-data/clk for SSI on APB.	RW
29	FTSSI	0: 0.5-data/clk for TSSI on APB;1: 1-data/clk for TSSI on APB.	RW
28	FUART	0: 0.5-data/clk for UART on APB;1: 1-data/clk for UART on APB.	RW
27	FAIC	0: 0.5-data/clk for AIC on APB;1: 1-data/clk for AIC on APB.	RW
26:4	Reserved	Write has no effect, read as zero.	R
3	HLT	Global halt status, halt occurs in any channel, the bit should be set to 1 by hardware and cleared to 0 by software. 0: no halt occurred 1: halt occurred in one or more channels	RW
2	AR	Global address error status, address error occurs in any channel, the bit should be set to 1 by hardware and cleared to 0 by software. 0: no address error occurred 1: address error occurred	RW
1	Reserved	Write has no effect, read as zero.	R
0	DMAE	Global DMA enable. 0: disable DMA 1: enable DMA	RW

NOTES:

1. any of FMSC/FSSI/FTSSI/FUART/FAIC has been set，the corresponding DMA transfer on APB for MSC(MSC1, MSC2), SSI(SSI1), UART0~4, AIC will be in fast mode.
2. DMAE is a global switch, so software must be careful to toggle it from 0 to 1 or 1 to 0. It is

UNPREDICTABLE to set 0 to DMAE abruptly when some channels are still working.

3. For more detail of CH01, refer to later chapter of special channel 0 and channel 1

16.6.2 DMA Interrupt Pending (DIRQP)

	DIRQP																													BASE+0x1004
Bit	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
	Reserved																								IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0
RST	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

Bits	Name	Description	RW
31:8	IRQn	Reserved, read as 0, write has no effort.	RW
7:0	IRQn	IRQn denotes pending IRQ to main CPU for corresponding channel. 0: normal DMA transfer is in-progress or the channel is idle 1: an address error occurs (only available for UARTn) or normal DMA transfer done; moreover, if the relative channel is programmable, MCU can only set 1 to it. Hardware can only set 1 to it and main CPU can only set 0 to it.	RW

16.6.3 DMA Doorbell (DDB)

	DDB																													BASE+0x 1008
Bit	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
	Reserved																								DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
RST	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

Bits	Name	Description	RW
31:8	Reserved	Write has no effect, read as zero.	R
7:0	DDBn	DMA Doorbell. Software sets it to 1 and hardware clears it to 0. 0: disable DMA controller to fetch the first descriptor or DMA controller clears it to 0 as soon as it starts to fetch the descriptor 1: Writing 1 to DDS will set the corresponding DDBn bit to 1 and enable DMA controller to fetch the first descriptor For example, write 0x00000001 to DDS, DDB0 bit is set to 1 and enable DMA channel 0 to fetch its first descriptor. Write 0 to DDS, no meaning.	RWS

16.6.4 DMA Doorbell Set (DDS)

	DDS																													BASE+0x 100C
Bit	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
	Reserved																								DDS7	DDS6	DDS5	DDS4	DDS3	DDS2	DDS1	DDS0
RST	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

Bits	Name	Description	RW
31:8	Reserved	Write has no effect, read as zero.	R
7:0	DDSn	DMA Doorbell Set for each channel. Read as zero, and when write: 0: ignore 1: Set the corresponding DDBn bit to 1	W

16.6.5 Descriptor Interrupt Pending (DIP)

	DIP																													BASE+0x 1010
Bit	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
	Reserved																								DIP7	DIP6	DIP5	DIP4	DIP3	DIP2	DIP1	DIP0
RST	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

Bits	Name	Description	RW
31:8	Reserved	Write has no effect, read as zero.	R
7:0	DIPn	Descriptor interrupt pending. If any of the descriptor, except the last one, set TIE, the corresponding bit of this register shall set after the very descriptor finished. This function provide more flexible method to be aware of the process of the long descriptor links. 0: No descriptor interrupt pending 1: The corresponding bit channel has completed a descriptor with TIE Read only for software..	R

16.6.6 Descriptor Interrupt Clear (DIC)

	DIC																													BASE+0x 1014
Bit	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

	Reserved	DIC7	DIC6	DIC5	DIC4	DIC3	DIC2	DIC1	DIC0
RST	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

Bits	Name	Description	RW
31:8	Reserved	Write has no effect, read as zero.	R
7:0	DICn	Descriptor interrupt pending clear. 1: ignore 0: Clear the corresponding DIPn bit Only write 1 for software, read has no effect.	W1

16.6.7 DMA Channel Programmable (DMACP)

	DMACP																BASE+0x 101C													
Bit	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
	DCP31	DCP30	DCP29	DCP28	DCP27	DCP26	DCP25	DCP24	DCP23	DCP22	DCP21	DCP20	DCP19	DCP18	DCP17	DCP16	DCP15	DCP14	DCP13	DCP12	DCP11	DCP10	DCP9	DCP8	DCP7	DCP6	DCP5	DCP4	DCP3	DCP2	DCP1	DCP0
RST	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

Bits	Name	Description	RW
31:0	DCPn	Channel programmable enable. 0, compatible with previous Ingenic SOC; 1, firmware controlled channel	RW

16.6.8 DMA Soft IRQ Pending (DSIRQP)

	DSIRQP																BASE+0x 1020													
Bit	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
	SIRQ31	SIRQ30	SIRQ29	SIRQ28	SIRQ27	SIRQ26	SIRQ25	SIRQ24	SIRQ23	SIRQ22	SIRQ21	SIRQ20	SIRQ19	SIRQ18	SIRQ17	SIRQ16	SIRQ15	SIRQ14	SIRQ13	SIRQ12	SIRQ11	SIRQ10	SIRQ9	SIRQ8	SIRQ7	SIRQ6	SIRQ5	SIRQ4	SIRQ3	SIRQ2	SIRQ1	SIRQ0
RST	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

Bits	Name	Description	RW
31:0	SIRQn	Soft IRQ to MCU. Set any bit of this register to 1, if not masked correspond with that in DSIRQM, will issue a software interrupt to MCU. Note that the register is read-only for other masters but is entirely controlled (R/W) by the MCU of PDMA	R

16.6.9 DMA Soft IRQ Mask (DSIRQM)

	DSIRQM																BASE+0x 1024													
Bit	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
	SIRQM31	SIRQM30	SIRQM29	SIRQM28	SIRQM27	SIRQM26	SIRQM25	SIRQM24	SIRQM23	SIRQM22	SIRQM21	SIRQM20	SIRQM19	SIRQM18	SIRQM17	SIRQM16	SIRQM15	SIRQM14	SIRQM13	SIRQM12	SIRQM11	SIRQM10	SIRQM9	SIRQM8	SIRQM7	SIRQM6	SIRQM5	SIRQM4	SIRQM3	SIRQM2	SIRQM1	SIRQM0
RST	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Name	Description	RW
31:0	SIRQMn	Soft IRQ mask. 0, not mask corresponding DSIRQP soft IRQ; 1, mask corresponding DSIRQP soft IRQ Note that the register is read-only for other masters but is entirely controlled (R/W) by the MCU of PDMA	R

16.6.10 DMA Channel IRQ Pending to MCU (DCIRQP)

	DCIRQP																BASE+0x 1028													
Bit	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
	Reserved																								CIRQ7	CIRQ6	CIRQ5	CIRQ4	CIRQ3	CIRQ2	CIRQ1	CIRQ0
RST	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

Bits	Name	Description	RW
31:8	Reserved	Write has no effect, read as zero.	R
7:0	CIRQn	Channel IRQ Pending to MCU. The register is read-only 0, no channel IRQ to MCU; 1, pending channel IRQ to MCU When channel n is not programmable, the CIRQn is ignored by hardware. When channel n is programmable and DCMn.TIE is active 1, DIRQP.IRQn can not be set automatically any more after address error occurs or TT becomes active 1, instead, if DCIRQMn==0, CIRQn may be set 1 immediately to raise an IRQ to MCU. Moreover, when an channel m with above attribute is bound with the INTC_IRQ, if DCIRQMm==0, an active INTC_IRQ request will trigger an active CIRQn immediately	R

16.6.11 DMA Channel IRQ to MCU Mask (DCIRQM)

	DCIRQM																BASE+0x 102C							
Bit	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
	Reserved																								CIRQM7	CIRQM6	CIRQM5	CIRQM4	CIRQM3	CIRQM2	CIRQM1	CIRQM0
RST	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	1

Bits	Name	Description	RW
31:8	Reserved	Write has no effect, read as zero.	R
7:0	CIRQMn	Mask of Channel IRQ to MCU. The register is read-only for other master but is entirely controlled (R/W) by the MCU of PDMA 0, not mask corresponding channel's IRQ to MCU; 1, mask corresponding channel's IRQ to MCU When channel n is not programmable, the CIRQMn is ignored by hardware. When channel n is programmable and DCMn.TIE is active 1, CIRQMn can mask corresponding channel's IRQ to MCU.	R

16.7 MCU

MCU in PDMA is a mini CPU compatible with XBurst-1 ISA without implementing CACHE, MMU, DEBUG, FPU and MXU. It is very similar as the AUX in VPU but has different memory mapped control and status register.

16.7.1 MCU Control & Status

	DMCS																BASE+0x 1030															
Bit	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
	SLEEP	SCMD	Reserved						SC _OFF																Reserved				SC_CALL	Reserved		SW_RST
RST	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	0	0	0	0	1

Bits	Name	Description	RW
31	SLEEP	Sleep status of MCU. Read only by main CPU.	R
30	SCMD	Security Mode. Read only by main CPU.	R
29:24	reserved	Write has no effect, read as zero.	R
23:8	SC_OFF	Set the offset of the caller's data structure for SC_CALL	RW
7:4	reserved	Write has no effect, read as zero.	R

3	SC_CALL	Security Call. Writing 1 triggers a security exception which directs MCU to run security code located in the on-chip security ROM., read as zero.	W
2:1	reserved	Write has no effect, read as zero.	R
0	SW_RST	Software reset. 1, MCU keeps at reset state; 0, do not soft-reset MCU any more	RW

NOTES:

BCH_DB, BCH_DF and BCH_EF are only available when special channel 0 and channel 1 is used for implementing storage of raw NAND chipsets. Please refer to BCH spec for detail.

16.7.2 MCU Normal MailBox

	DMNMB																BASE+0x 1034															
Bit	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
	MB																															
RST	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

Bits	Name	Description	RW
31:0	MB	Contents of Normal Mailbox. Writing any value to the register will trigger a normal mailbox IRQ to main CPU when relative MASK field in DMINT is not set	RW

16.7.3 MCU Security MailBox

	DMSMB																BASE+0x 1038															
Bit	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
	MB																															
RST	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

Bits	Name	Description	RW
31:0	MB	Contents of Security Mailbox. Writing any value to the register will trigger a security mailbox IRQ to main CPU when relative MASK field in DMINT is not set. Note that writing the register is inhibited by HW when DMCS.SCMD == 0	RW

16.7.4 MCU Interrupt

	DMINT																BASE+0x 103C													
Bit	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
	Reserved														S_IP	N_IP	Reserved														S_IMSK	N_IMSK
RST	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	1

Bits	Name	Description	RW
31:18	Reserved	Write has no effect, read as zero.	R
17	S_IP	Security Mailbox IRQ pending. 1-pending IRQ. Hardware only set 1 by writing DMSMB when DMCS.SCMD == 1. Can be clear to 0 in any mode.	RW
16	N_IP	Normal Mailbox IRQ pending. 1-pending IRQ. Hardware only set 1 by writing DMNMB. Can be clear to 0 in any mode.	RW
15:2	Reserved	Write has no effect, read as zero.	R
1	S_IMSK	Security Mailbox IRQ mask. 1-mask; 0-not mask Can only be modified in security mode.	RW
0	N_IMSK	Normal Mailbox IRQ mask. 1-mask; 0-not mask	RW

16.7.5 Multiple Bank Tightly Coupled Sharing Memory

In MCU, there is a Tightly Coupled Sharing Memory (TCSM) with 8KB size and total 2 banks. Multiple- bank TCSM permits several masters accessing different banks in parallel. TCSM works at the same speed as MCU and both instruction fetch and data read/write to TCSM by MCU can be performed in one cycle.

Table 16-7 TCSM space

	VA (only available for MCU)	PA (not remapped)	PA (remapped)
BANK0	F4000000h ~ F4000FFFh	13422000h ~ 13422FFFh	F3422000h ~ F3422FFFh
BANK1	F4001000h ~ F4001FFFh	13423000h ~ 13423FFFh	F3423000h ~ F3423FFFh

16.7.6 CP0 Registers of MCU

Available CP0 registers of MCU are not entirely compatible with MIPS PRA spec, they are listed below.

CP0 Name (number)	Description	*sIt*
STATUS (12)	Bit 31~27:: reserved, read as zero Bit26: readable; only 1 can bet written into*1	*0*

	Bit25: readable; only 1 can bet written into*1 Bit24: only being active during running Security Call routine *2 Bit2: ERL, Bit1: EXL, Bit0: IE	
INTCTL (12)	Bit31~3: reserved, read as zero; Bit2: IRQ pending of soft IRQ to MCU; Bit1: IRQ pending of channel IRQ to MCU; Bit0: IRQ pending of INTC_IRQ Note that the register is read-only by MCU	*1*
CAUSE (13)	Bit31:BD， whether the interrupted PC is located in branch delay slot Bit30 ~ 0: reserved, read as zero	*0*
SeCAUSE (13)	Bit31:BD, similar as CAUSE, but being active when an IRQ interrupted a running Security Call routine. Bit30 ~ 0: reserved, read as zero	*1*
EPC (14)	IRQ Exception program counter	*0*
SePC (24)	Security Exception program counter	*0*
ErrorPC (30)	*Reset exception program counter*	*0*

NOTE:
*1 : once value 1 is set, only chip reset signal can clear it to 0.
*2 : STATUS.Bit24 is hard wired outside of PDMA.

16.7.7 Normal Exceptions Accepted by MCU

There are two kinds of normal exceptions that can be accepted by MCU: RESET and IRQ.
Note that RESET exception has the handler entry: 0xF4000000, while IRQ has another one: 0xF4000100.
Once the MCU accepts a normal exception, according to the definition of XBurst-1 Programming Manual, current interrupted PC will be pushed into CP0.EPC or CP0.ErrPC, and further CP0.STATUS.ERL or CP0.SATUS.EXL will be automatically set 1. Firmware can take use of MTC0/MFC0 to access these three CP0 registers: EPC, ErrPC and STATUS. Firmware must use ERET instruction to return to the interrupted locale to resume executing.

16.7.8 How to Boot MCU Up

After power-on reset, MCU keeps at reset state due to DMCS.SW_RST==1. To boot MCU up, first of all, RESET exception handler must be loaded into TCSM at the physical address range beginning from 0x13422000 or 0xF3422000 for remapped case. Programmer can use main CPU or some Non-programmable DMA channel to finish this loading work. Then clear DMCS.SW_RST to 0 by main CPU so that MCU can acknowledge a RESET exception, now MCU begins to run from the PC 0xF4000000. Following is a simple example.

1. Prepare a simple RESET exception handler, total 3 instructions:

```
1:  WAIT    //sleep
B   1b      //endless loop
```

NOP //delay slot

2. Load the handler into TCSM at the physical address range beginning from 0x13422000 or 0xF3422000 for remapped case.
3. Clear DMCS.SW_RST to 0 by main CPU

Now MCU will execute WAIT instruction and then enter sleep state

16.8 DMA manipulation

16.8.1 Descriptor Transfer Mode

16.8.1.1 Non-Stride (1-Dimensional) Transfer Mode

To do proper Descriptor DMA transfer in a channel, do following steps:

1 Check whether the status of DMA controller is available, that is, for global control (DMAC), ensure that DMAC.AR=0 and DMAC.HLT=0; and then for expected channel n, ensure that DCSn.AR=0, DCSn.HLT=0, DCSn.TT=0, DTCn=0.

2 guarantee DCSn.NDES=0, and select 4 word or 8 word descriptor by DCSn.DES8.

Build descriptor in memory. Write the first descriptor's address in DDAn and the address must be 16Bytes aligned for 4word descriptor and 32Bytes aligned for 8word descriptor, otherwise, the hardware behavior later is UNPREDICTABLE**. The address includes two parts: Base and Offset address. If the descriptor is linked, the 32-bit address of next descriptor is composed of 20-bit Base address in DDAn and 8-bit Offset address in DESn.CDOA. See**

3 Table 16-8 for the detailed descriptor structure.

4 Don't update DRTn manually by software, instead, hardware will manage the request type of descriptor's transfer automatically.

5 Set 1 to the corresponding bit in DDB to initiate descriptor fetch.

6 Set DMAC.DMAE=1 and expected DCSn.CTE=1 to activate channel n.

7 Hardware will automatically clear the corresponding bit in DDB as soon as it starts to fetch the descriptor.

8 Wait for DMA request from peripherals to start channel n's DMA transfer.

9 After DMA transfers described by the current descriptor complete, if current DCMn.LINK =0, then DCSn.TT will be set to 1 immediately. Further, if DCMn.TIE=1, it will issue an interrupt request. However, if DCMn.LINK=1, next descriptor will be automatically fetched to repeat step 8 and 9 but ignore setting DCSn.TT until a final descriptor with settting of DCMn.LINK=0 has been done.

10 When channel n's transfer ends normally (DCSn.TT=1) or abnormally (DCSn.HLT=1), to correctly reuse the channel later, software must close the channel first by setting 0 to DCSn.CTE, and then safely clear DCSn.TT or DCSn.HLT to 0.

Table 16-8 Descriptor Structure

Word	Bit	Name	Function
1st (DES0)	31-24	Reserved	--
	23	SAI	Source Address Increment
	22	DAI	Target Address Increment
	21-20	Reserved	--
	19-16	RDIL	Request Detection Interval Length
	15-14	SP	Source port width
	13-12	DP	Target port width
	11	Reserved	--
	10-8	TSZ	Transfer Data Size
	7-3	Reserved	--
	2	STDE	Stride transfer enable
	1	TIE	Transfer Interrupt Enable
	0	LINK	Descriptor Link Enable
2nd (DES1)	31-0	DSA	Source Address
3rd (DES2)	31-0	DTA	Target Address
4th (DES3)	31-24	DOA	Descriptor Offset address
	23-0	DTC	Transfer Counter
5th (DES4)	31-16	TSD	Target Stride Address
	15-0	SSD	Source Stride Address
6th(DES5)	31-6	Reserved	--
	5-0	DRT	DMA Request Type
7th(DES6)	31-0	Reserved	--
8th(DES7)	31-0	Reserved	--

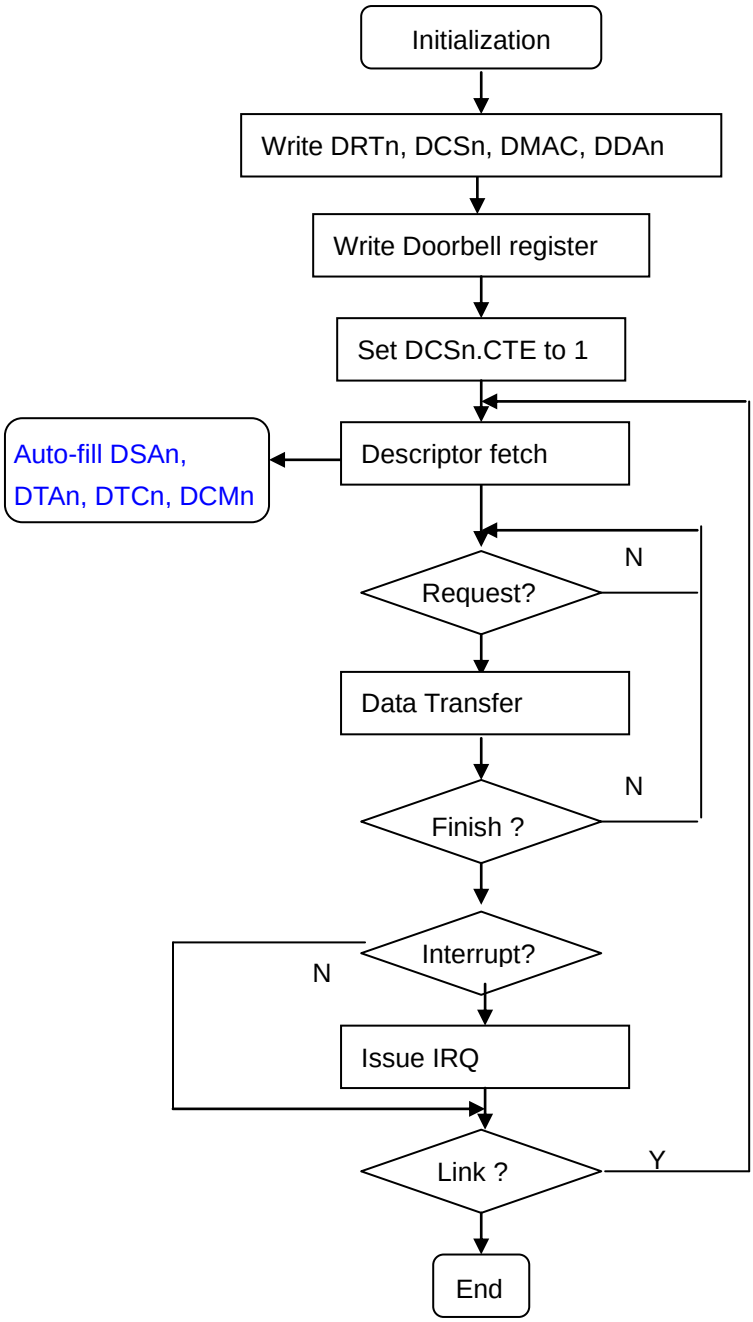


Figure 16-2 Descriptor Transfer Flow

16.8.1.2 Stride (2-dimensional) Transfer Mode

Stride transfer mode means total transfer counts are subdivided into dozens of or even hundreds of sub-blocks with same size, and each adjacent two sub-blocks has fixed nonzero address gap between the tail of preceding one and the head of subsequent one.

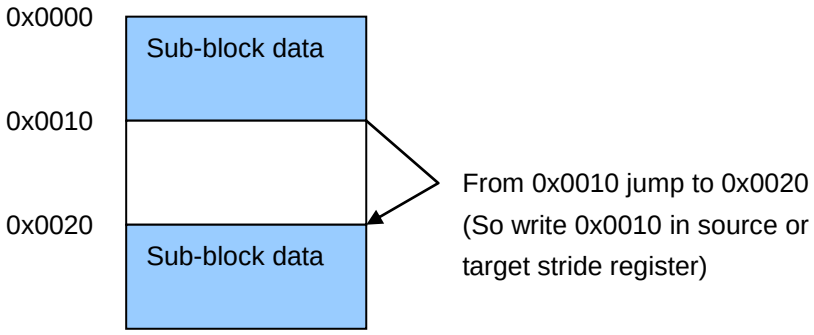


Figure 16-3 Example for Stride Transfer Mode

Manipulation steps of Stride Transfer Mode are similar to Non-stride Transfer Mode. However, each new sub-block's start address (source or destination) is performed by adding the next adjacent byte address of very completed sub-block's end address to corresponding stride difference.

16.8.2 No-Descriptor Transfer Mode

To do proper No-Descriptor DMA transfer, do following steps:

1 Check whether the status of DMA controller is available, that is, for global control (DMAC), ensure that DMAC.AR=0 and DMAC.HLT=0; and then for expected channel n, ensure that DCSn.AR=0, DCSn.HLT=0, DCSn.TT=0, DTCn=0.
2 Forx channel n, initialize DSAn, DTAn, DTCn, DRTn, DCSn, DCMn properly.
3 Set DMAC.DMAE=1 and DCSn.CTE=1 to launch DMA transfer.

For a channel with auto-request (DRTn.RT=0x8), the transfer begins automatically when the DCSn.CTE bit and DMAC.DMAE bit are set to 1. While for a channel with other request types, the transfer does not start until a transfer request is issued and detected.

For any channel n, The DTCn value is decremented by 1 for each successful transaction of a data unit. When the specified number of transfer data unit has been completed (DTCn = 0), the transfer ends normally. Meanwhile if DCMn.TIE=1, corresponding bit of DIRQP will be set to 1 to raise an interrupt request. However, during the transfer, if a DMA address error occurs, the transfer should be suspended, both DCSn.AR and DMAC.AR are set to 1 as well as corresponding bit of DIRQP to raise an interrupt request despite of DCMn.TIE.

Sometimes, for example, an UART parity error occurs for a channel that is transferring data between such UART and another terminal. In the case, both DCSn.HLT and DMAC.HLT are set to 1 and the transfer is suspended. Software should identify halt status by checking such two bits and re-configure DMA to let DMA rerun properly later.

16.8.3 Descriptor Transfer Interrupt/Stop control

DMA descriptor provides a more fast and easy way to transfer data. Usually in a long descriptor

linked-line, it's hard to specifically know where the current descriptor point is. Here are the recommended ways to help software to control this.

Register	Part	Name	Function
DCSn	15:8	CDOA	Copy of offset address of last completed descriptor from that in DMA command register.
	0:0	CTE	Channel transfer enable: 0: disable; 1: enable. **Note: If set during a long descriptor linked-line, DMA will finish the current data unit transfer and terminate the whole transfer. If it is not necessary, don't use this method to end the transfer.**
DCMn	1:1	TIE	Transfer Interrupt Enable (TIE). 0: disable interrupt; 1: enable interrupt when TT is set to 1. **Note: If set during a long descriptor linked-line, DMA will raise a interrupt to INTC, if not masked, and this can be acknowledged by visit DSP register. Also if TIE has already been set in the descriptor, DMA will also raise interrupt except that is the last descriptor, and if it happens, DIRQP will raise instead of DIP.**
	0:0	LINK	Descriptor Link Enable. 0: disable; 1: enable. **Note: If set during a long descriptor linked-line, DMA will finish the current descriptor, and then regard it as a no-linked descriptor and stop transfer. However, it has a change that write this bit to 0 is ignored by hardware, so a double write or a read-check is recommended.**
DIP	7:0	DIP	Descriptor interrupt pending. If any of the descriptor, except the last one, set TIE, the corresponding bit of this register shall set after the very descriptor finished. This function provide more flexible method to be aware of the process of the long descriptor links. 0: No descriptor interrupt pending 1: The corresponding bit channel has completed a descriptor with TIE Read only for software. **Note: DIP will raise if the two conditions satisfy, one is this is not the last descriptor, the other is the current TIE in DCMn or TIE in current descriptor is set.**

16.9 DMA Requests

DMA transfer requests are normally generated from either the data transfer source or target, but also they can be issued by on-chip peripherals that are neither the source nor the target. There are two DMA transfer request types: auto-request, and on-chip peripheral request. For any channel n, its transfer request type is determined through DRTn.

16.9.1 Auto Request

When there is no explicit transfer request signal available, for example, memory-to-memory transfer or memory to some on-chip peripherals like GPIO, the auto-request mode allows the DMA to automatically generate a transfer request signal internally. Therefore, when DMA initialization done, once the DMAC.DMAE and DCSn.CTE are set to 1, the transfer begins immediately in channel n which DRTn=0x8.

16.9.2 On-Chip Peripheral Request

In the mode, transfer request signals come from on-chip peripherals or even off-chip devices. When a device issues a peripheral request to its corresponding channel n, the transferred byte number for the request is equals to:

1. byte number determined by the bytes denoted by DCMn.TSZ when DCMn.TSZ != 7
2. byte number determined by the bytes denoted by RDIL when DCMn.TSZ == 7

Be careful that the above number must not exceed the device's expected one otherwise FIFO under-run or over-run may occur. However, if the number is too small than the expected one, DMA's bus efficiency will become worse.

16.10 How to Use Programmable DMA Channel

When system needs to setup one programmable channel for later use of smart data transfer, do following steps:

1. Load elaborate firmware to TCSM, should use physical address 0x13422xxx. Common code fragment of IRQ exception handler must be located at physical address range beginning from 0x13422100.
2. Prepare a DMA channel n using expected transfer mode as description in preceding chapters
3. Must set DCMn.TIE=1 and DMACP.DCPn=1
4. Must ensure DCIRQM.CIRQMn==0
5. Launch the channel n as description in preceding chapters

Now the channel n becomes a working programmable channel. It is the responsibility of local channel IRQ's handler to remove corresponding channel IRQ source by setting 0 to DCSn.TT.

Moreover, if programmer wants to let the channel can be triggered by local soft IRQ, DSIRQM.SIRQMn must be set 0, then at expected moment, the MCU sets 1 to DSIRQP.SIRQn to trigger a soft IRQ. It is the responsibility of this soft IRQ's handler to remove the IRQ by setting 0 to DSIRQP.SIRQn.

17 Real Time Clock

17.1 Overview

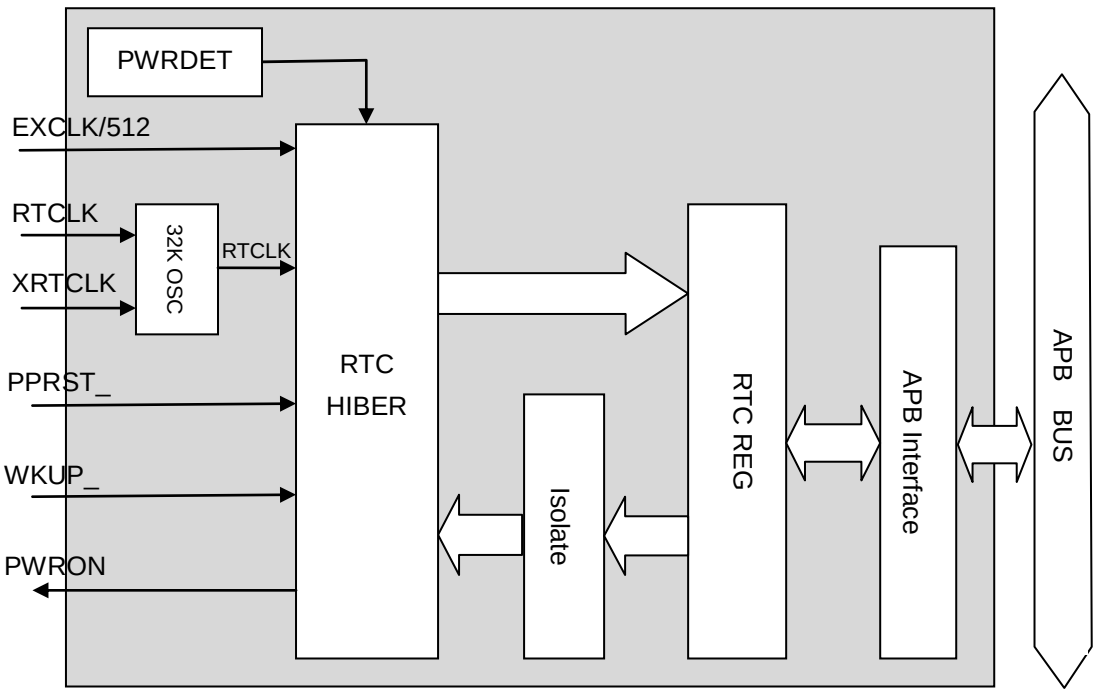
The Real-Time Clock (RTC) unit can be operated in either chip main power is on or the main power is down but the RTC power is still on. In this case, the RTC power domain consumes only a few micro watts power.

The RTC contains a real time, alarm logic and the power down and wakeup control logic.

17.2 Features

- Embedded 32768Hz oscillator for 32k clock generation with an external 32k crystal
- RTCLK selectable from the oscillator or from the divided clock of EXCLK, so that 32k crystal can be absent if the hibernating mode is not needed
- 32-bits second counter
- Programmable and adjustable counter to generate accurate 1 Hz clock
- Alarm interrupt, 1Hz interrupt
- Stand alone power supply, work in hibernating mode
- Power down controller
- Alarm wakeup
- External pin wakeup with up to 2s glitch filter

17.3 Block Diagram



17.4 Pins Description

RTC has 6 signal IO pins and 1 power pin. They are listed and described in.

Pin Names	Pin Loc	IO	IO Cell Char.	Pin Description	Power
RTCLK		AI	32768Hz	RTCLK: 32768 clock input or OSC input	VDDRTC33
XRTCLK		AO		XRTCLK: OSC output	VDDRTC33
PWRON		O		PWRON: Power on/off control of main power	VDDRTC33
WKUP_		I	Schmitt	WKUP_: Wake signal after main power down	VDDRTC33
PPRST_		I	Schmitt	PPRST_: RTC power on reset and RESET-KEY reset input	VDDRTC33
VDDRTC33		P		VDDRTC33: 3.3V power for RTC and hibernating mode controlling that never power down (normally you can use 1.8V instead to reduce power consumption)	-
LDOOUT		AIO		LDOOUT: capacitor pin for RTC LDO, need a 1nF decoupling capacitor to ground	-

RTCLK pin. need input 32.768KHz rtc clock. If do not use any clock, hibernate mode will be NOT available any more, and the time will lose if power down.

PWRON pin: this pin is used to control the main power on/off. Output high voltage means on and low voltage means off.

WKUP_ pin: hibernating mode wakeup input. (low active)

PPRST_ pin: This pin should be set to low voltage only in two cases.

- When RTC power is turned on. (so that whole chip is power on)
- A RESET-KEY is pressed.

17.5 Registers Description

Following chapter will descript the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. The registers can be read and written by APB bus
3. Register read/write attribute

   R - Read only

   W - Write only

   RW - read and write

   RCW - read and write, but clear to 0 by read

   RSW - read and write, but set to 1 by read

   RWC - read and write, clear to 0 by write 1, write 0 has no effect

RWS - read and write, set to 1 by write 1, write 0 has no effect

RC - read only, and clear to 0 by read

RS - read only, and set to 1 by read

SPEC - special access method, relate to its description

4. Reset Value

1 - reset to 1

0 - reset to 0

? - value unknown after reset

Table 17-1 Registers Memory Map-Address Base

Name	Base	Description
RTC	0x10003000	Address base of RTC

Table 17-2 Registers for real time clock

Name	Description	Reset Value	Address offset	Access Size
RTCCR	RTC Control Register	0x000000??*1*2	0x000	32
RTCSR	RTC Second Register	0x????????	0x004	32
RTCSAR	RTC Second Alarm Register	0x????????	0x008	32
RTCGR	RTC Regulator Register	0x0???????	0x00C	32

NOTES:

1 *1: Unless otherwise stated, the reset value is for PPRST_ and Hibernating wakeup reset. WDT reset doesn't change the value.

2 *2: The reset value can be either of 0x00000081, 0x00000091, 0x00000089, 0x00000099.

Table 17-3 Registers for hibernating mode

Name	Description	Reset Value	Address offset	Access Size
HCR	Hibernate Control Register	0x00000000*1	0x020	32
HWFCR	Wakeup filter counter Register in Hibernate mode	0x0000???0	0x024	32
HRCR	Hibernate reset counter Register in Hibernate mode	0x00000??0	0x028	32
HWCR	Wakeup control Register in Hibernate mode	0x2d52d2d0*1	0x02C	32
HWRSR	Wakeup Status Register in Hibernate mode	0x00000???*1	0x030	32
HSPR	Scratch pattern register	0x????????	0x034	32
WENR	Write enable pattern register	0x00000000	0x03C	32

WKUPPINCR	Configure the extend press WKUP pin to reset chip	0x00050064	0x048	32

NOTE:

*[1]: Unless otherwise stated, the reset value is for PPRST_ and Hibernating wakeup reset. WDT reset doesn't change the value.

All these registers, include those for real time clock and for hibernating mode control, except otherwise stated, are implemented in RTCLK clock domain. When write to these registers, it needs about 1 ~ 2 RTCLK cycles to actually change the register's value and needs another RTCLK cycle to allow the next write access. A bit RTCCR.WRDY is used to indicate it. When RCR.WRDY is 1, it means the previous write is finished, a right value can be read from the target register, and a new write access can be issued. So before any write access, please make sure RCR.WRDY = 1.

17.5.1 RTC Control Register (RTCCR)

RTCCR contains bits to configure the real time clock features. Unless otherwise stated, the reset value is for PPRST_ and Hibernating wakeup reset. WDT reset doesn't change the value.

	RTCCR																BASE + 0x000													
Bit	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
	Reserved																								WRDY	1HZ	1HZIE	AF	AIE	AE	SELEXC	RTCE
RST	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]	0	0*[1]	?	0	?	0*[1]	1

NOTE:

*[1]: These bits are reset in all resets: PPRST_ input pin reset, hibernating reset and WDT reset.

Bits	Name	Description	RW
31:7	Reserved	Writing has no effect, read as zero.	R
7	WRDY	Write ready flag. It is 0 when a write is currently processing and the value has not been written to the writing target register. No write to any RTC registers can be issued in this case, or the result is undefined. The read value from the target register is also undefined. The reading is meaningful and another write can be issued when it is 1. This bit is read only and write to it is ignored.	R
6	1HZ	1Hz flag. This bit is set by hardware once every 1 second through the 1Hz pulse if the real time clock is enabled (RTCCR.RTCE = 1). This bit can be cleared by software. Write 1 to this bit is ignored.	RW
5	1HZIE	1Hz interrupt enable. Writing to this bit takes effect immediately without delay.	RW

		1HZIE: 0 — 1Hz interrupt is disabled. 1 — 1Hz interrupt is enabled. RTC issues interrupt when 1HZ bit is set.	
4	AF	Alarm flag. This bit is set by hardware when alarm match (RTCSR = RTCSAR) is found and alarm is enabled (RTCCR.AE = 1) and the real time clock is enabled (RTCCR.RTCE = 1). This bit can be cleared by software. Write 1 to this bit is ignored. Writing to this bit takes effect immediately.	RW
3	AIE	Alarm interrupt enable. AIE: 0 — Alarm interrupt is disabled. 1 — Alarm interrupt is enabled. RTC issues interrupt when AF is set.	RW
2	AE	Alarm enable. AE: 0 — Alarm function is disabled. 1 — Alarm function is enabled.	RW
1	SELEXC	The divided EXCLK is selected as RTCLK in rtc-hiber module. SELEXC: 0 — OSC32K or RTCLK input clock is selected as RTCLK in rtc-hiber module. 1 — The divided EXCLK is selected as RTCLK in rtc -hiber module. NOTE: If do not use any 32Khz clock (either input clock or using crystal), hibernate mode will be NOT available any more, and the time will lose if power down. CPM.OPCR.ERCS must be 0, when using SELEXC = 1. When the main chip power down, SELEXC will be 0 in internal circuit, in this time, RTCLK will use OSC32K clock.	RW
0	RTCE	Real time clock enable. RTCE: 0 — Real time clock function is disabled. 1 — Real time clock function is enabled.	RW

17.5.2 RTC Second Register (RTCSR)

RTCSR is a 32-bit width second counter. It can be read and write by software. It is increased by 1 at every 1Hz pulse if the real time clock is enabled (RTCCR.RTCE = 1). When read, it should be read continued more than once and take the value if the adjacent results are the same. RTCSR is not initialized by any reset.

	RTCSR																														BASE + 0x004	
Bit	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
	RTCSR																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

17.5.3 RTC Second Alarm Register (RTCSAR)

RTCSAR serves as a second alarm register. Alarm flag (RTCCR.AF) is set to 1 when the RTCSR equals the RTCSAR in the condition of alarm is enabled (RTCCR.AE = 1) and the real time clock is enabled (RTCCR.RTCE = 1). RTCSAR can be read and write by software and is not initialized by any reset.

	RTCSAR																														BASE + 0x008	
Bit	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
	RTCSAR																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

17.5.4 RTC Regulator Register (RTCGR)

RTCGR is serves as the real time clock regulator, which is used to adjust the interval of the 1Hz pulse.

	RTCGR																														BASE + 0x00C	
Bit	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
	LOCK	Reserved					ADJC										NC1HZ															
RST	0*[1]	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

NOTE:

*[1]: This bit is reset in all resets: PPRST_ input pin reset, hibernating reset and WDT reset.

Bits	Name	Description	RW
31	LOCK	Lock bit. This bit is used to safeguard the validity of the data written into the RTCGR register. Once it is set, write to RTCGR is ignored. This bit can only be set by software and cleared by (any type of) resets. **LOCK** / **Description** 0 / Write to RTCGR is allowed. 1 / Write to RTCGR is forbidden.	RW

30:26	Reserved	Writing has no effect, read as zero.	R
25:16	ADJC	This field specifies how many times it needs to add one 32kHz cycle for the 1Hz pulse interval in every 1024 1Hz pulses. In other words, among every 1024 1Hz pulses, ADJC number of them are trigged in every (NC1HZ + 2) 32kHz clock cycles, (1024 – ADJC) number of them are trigged in every (NC1HZ + 1) 32kHz clock cycles.	RW
15:0	NC1HZ	This field specifies the number plus 1 of the working 32kHz clock cycles are contained in the 1Hz pulse interval. In other words, 1Hz pulse is trigged every (NC1HZ + 1) 32kHz clock cycles, if RTCGR.ADJC = 0.	RW

17.5.5 Hibernate Control Register (HCR)

HCR contains the bit to control the main chip power on/off. Unless otherwise stated, the reset value is for PPRST_ and Hibernating wakeup reset.

	HCR																BASE + 0x020													
Bit	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
	Reserved																														PD	
RST	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

Bits	Name	Description	RW
31:1	Reserved	Writing has no effect, read as zero.	R
0	PD	Power down or power on bit. Besides writing by CPU, this bit will be set to 1 if an unknown reason main power supply off is detected. This bit controls the PWRON pin level. When co-working with some external components, this bit is used for power management of this chip. It is supposed when 1 is written to this bit, the main power supply of the chip, except RTC power, will be shut down immediately. After this bit is set to 1, all registers in RTC module, except RTCCR.1HZ and RTCCR.1HZIE, cannot be changed by write access. This bit is cleared by reset pin reset and hibernating reset. The later one is asserted by wakeup procedure.	RW

PD	PWRON	Description
0	VDDRTC33	No power down, keep power on.
1	0 V	Power down enable, turn power off.

17.5.6 HIBERNATE mode Wakeup Filter Counter Register (HWFCR)

The HIBERNATE mode Wakeup Filter Counter Register (HWFCR) is a 32-bit read/write register .It filter the glitch generated by a dedicated wakeup pin. The HWFCR is not initialized by any reset.

	HWFCR																													BASE + 0x024				
Bit	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
	Reserved																HWFCR											Reserved				
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	0	0	0	0	0

Bits	Name	Description	RW
31:16	Reserved	Writing has no effect, read as zero.	R
15:5	HWFCR	Wakeup pin effective minimum time in number of 32 RTCLK cycles, used as glitch filter logic. Maximum of 2 seconds if the RTCLK is 32768Hz If this value is configured to 0, and the WKUP pin keeps low longer than 15 RTCLK periods, it will wakes up RTC from Hibernate. The glitch filter time equals to HWFCR*32 RTCLK period.	RW
4:0	Reserved	Writing has no effect, read as zero.	R

17.5.7 Hibernate Reset Counter Register (HRCR)

The Hibernate Reset Counter Register is a 32-bit read/write register that specifies hibernate reset assertion time. The HRCR is initialized by PPRST_ .

	HRCR																													BASE + 0x028				
Bit	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
	Reserved																HRCR					Reserved											
RST	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	0	0	0

Bits	Name	Description	RW
31:15	Reserved	Writing has no effect, read as zero.	R
14:11	HRCR	HIBERNATE Reset asserting time. Number of 2048 RTCLK cycles. Default value 0x1, assert 125 ms. Maximum 1 second if the RTCLK is 32768Hz. If this value is configured to 0, it will generate 2048 RTCLK HIBERNATE Reset, about 62.5 milliseconds.	RW
10:0	Reserved	Writing has no effect, read as zero.	R

17.5.8 HIBERNATE Wakeup Control Register (HWCR)

The HIBERNATE Wakeup Control Register is a 32-bit read/write register that controls real time clock alarm wake up enable. The reset value only for PPRST_ .

	HWCR																BASE + 0x02C															
Bit	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
	EPDET																													Reserved		EALM
RST	0	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	0	0*[1]

NOTE:

*[1]: This bit is reset in PPRST_ input pin reset and hibernating reset .

Bits	Name	Description	RW
31:3	EPDET	Power detect enable. 0x5aa5a5a: enable(default) 0x1a55a5a5: disable Only 0x1a55a5a5 can disable power detect, other values will enable the power detect.	RW
2:1	Reserved	Writing has no effect, read as zero.	R
0	EALM	RTC Alarm wakeup enable. 0: disable 1: enable	RW

17.5.9 HIBERNATE Wakeup Status Register (HWRSR)

The HIBERNATE Wakeup Status Register is a 32-bit read/write register that reflects wakeup status bits.

	HWRSR																BASE + 0x030															
Bit	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
	Reserved																							APD	Reserved		HR	PPR	Reserved		PIN	ALM
RST	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*[2]	1*[1]	0	0	?	?

NOTES:

*[1]: This reset value only for PPRST_. It is undefined in case of other resets.

*[2]: This reset value only for HRST_. It is undefined in case of other resets.

Bits	Name	Description	RW
31:9	Reserved	Writing has no effect, read as zero.	R
8	APD	Accident power down. When the software has not set to HIBERNATE state, the core power is down, then an accident power down is detected. APD is set and remains set until software clears it. This bit can only be	RW

<table>
<tr><td></td><td></td><td>written with 0. Write with 1 is ignored.<table><tr><th>HR</th><th>Description</th></tr><tr><td>0</td><td>Accident power down has not occurred since the last time the software clears this bit.</td></tr><tr><td>1</td><td>Accident power down has occurred since the last time the software clears this bit.</td></tr></table></td><td></td></tr>
<tr><td>7:6</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>5</td><td>HR</td><td>Hibernate Reset. When a Hibernate reset detected, HR is set and remains set until software clears it. This bit can only be written with 0. Write with 1 is ignored.<table><tr><th>HR</th><th>Description</th></tr><tr><td>0</td><td>Hibernate reset has not occurred since the last time the software clears this bit.</td></tr><tr><td>1</td><td>Hibernate reset has occurred since the last time the software clears this bit.</td></tr></table></td><td>RW</td></tr>
<tr><td>4</td><td>PPR</td><td>PAD PIN Reset. When a PPRST_ is detected, PPR is set and remains set until software clears it. This bit can only be written with 0. Write with 1 is ignored.<table><tr><th>PPR</th><th>Description</th></tr><tr><td>0</td><td>PPRST_ reset has not occurred since last time the software clears this bit.</td></tr><tr><td>1</td><td>PPRST_ reset has occurred since last time the software clears this bit.</td></tr></table></td><td>RW</td></tr>
<tr><td>3:2</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>1</td><td>PIN</td><td>Wakeup Pin Status bit. The bit is cleared when chip enters hibernating mode. It is set when exit the hibernating mode by wakeup pin. This bit can only be written with 0. Write with 1 is ignored.</td><td>RW</td></tr>
<tr><td>0</td><td>ALM</td><td>RTC Alarm Status bit. The bit is cleared when chip enters hibernating mode. It is set when exit the hibernating mode by alarm. This bit can only be written with 0. Write with 1 is ignored.</td><td>RW</td></tr>
</table>

17.5.10 Hibernate Scratch Pattern Register (HSPR)

This is a scratch register used to hold a pattern. The software can check the pattern is kept to know whether RTC power has ever been down and whether it is needed to setup the real time clock.

	HSPR																BASE + 0x034														
Bit	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
	PAT																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:0	PAT	The pattern.	RW

17.5.11 Write Enable Pattern Register (WENR)

This is a write protect register to may the rtc registers can write only in certain condition.

	WENR																BASE + 0x03C													
Bit	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
	WEN	Reserved															WENPAT															
RST	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31	WEN	The write enable flag. If the WENPAT is 0xA55A then this bit will be 1. When the WEN changes to 1, the RTCCR, RTCSR, RTCSAR, RTCGR, HCR, HWFCR, HRCR, HWCR, HWRSR, HSPR registers could be changed. But RTCCR.SELEXC, RTCCR.HZIE, RTCCR.WRDY may change in any time. This bit is read only and write to it is ignored. This bit only reset by PPRST_ and HRST_. There is an exception, when system does NOT have RTC 32Khz crystal. MUST write 1 to RTCCR.SELEXC before write to any value to any other registers. **WEN / Description** 0 – Other RTC registers is locked, write these registers will be ignored. 1 – Other RTC registers can be changed.	R
30:16	Reserved	Writing has no effect, read as zero.	R
15:0	WENPAT	The write enable pattern. Before writing any value to RTCCR, RTCSR, RTCSAR, RTCGR, HCR, HWFCR, HRCR, HWCR, HWRSR, HSPR registers, write 0xA55A to WENPAT to set these register writable. If this value is ok, WEN will change to 1. But RTCCR.SELEXC and RTCCR.HZIE are writable in any time. These bits are write-only, always read as 0.	W

17.5.12 WKUP_PIN_RST control register (WKUPPINCR)

This is a register used to control the enable and set the judge time to reset the chip when extend press the WKUP pin. The are only initialized by PPRST_ and HRST_.

	WKUPPINCR																BASE + 0x048													
Bit	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
	Reserved													OSE_EN	BIAS_CTRL		Reserved								P_JUD_LEN				P_RST_EN			
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0

Bits	Name	Description	RW
31:19	Reserved	Writing has no effect, read as zero.	R
18	OSC_EN	RTC internal oscillator enable 1: enable 0: disable	RW
17:16	BIAS_CTRL	Reserved, need keep the default value(2'h1).	RW
15:5	Reserved	Writing has no effect, read as zero.	R
7:4	P_JUD_LEN	WKUP up pin extend press length judge value The judge time equals to FLT_LEN multiply RTC 1HZ period, and only the active press length equal or greater than judge time, the chip reset can assert.	RW
3: 0	P_RST_EN	Enable reset chip by extend press WKUP pin Others value: disable 4'hB: enable	RW

17.6 Operation Flow

17.6.1 Registers Access

17.6.1.1 Registers Read

All registers can read directly, but the RTCSR and RTCSAR should be read continued more than once and take the value if the adjacent results are the same. Please check and make sure RTCCR.WRDY equals to 1 before execute a register read.

17.6.1.2 Registers Write

Only the RTCCR.SELEXC, RTCCR.HZIE, RTCGR.LOCK, WENR.WENPAT bits can write directly. Others bits need following the below procedure

1) Check RTCCR.WRDY, make sure it equals to 1
2) Write 0xa55a to WENR register to enable other registers write.
3) Waiting WENR.WEN change to 1
4) Check RTCCR.WRDY, make sure it equals to 1
5) Write the target register
6) Waiting RTCCR.WRDY equals to 1 to make sure the current write processing was finished.

17.6.2 Normal Mode

17.6.2.1 Power Detect

When PPRST_ reset, the HWCR.EPDET set to 0x5aa5a5a, and the power detect is enable. If you want To disable the power detect, you should set HWCR.EPDET to 0x1a55a5a5. If power detect is enable, when CORE power supply was lose, the RTC will into hibernate mode. And if power detect is not enable, the RTC can't enter hibernate mode even CORE power supply was lose.

17.6.2.2 Power On Timing Diagram

The RTC power on procedure like diagram 1 -1.

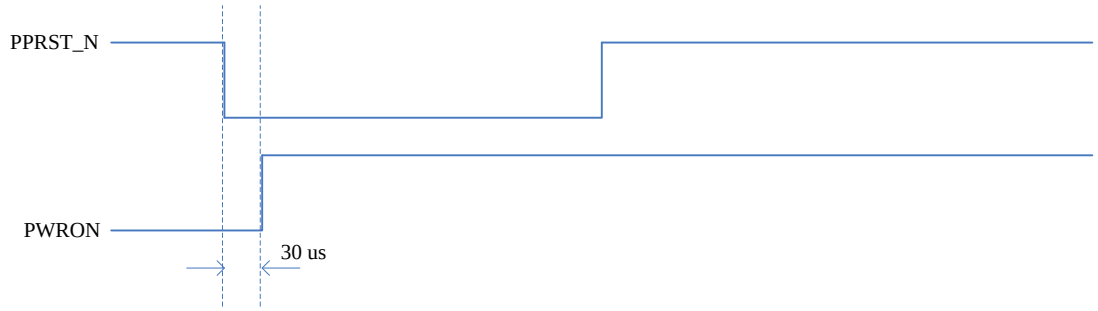


Figure 17-1 Core Power On

17.6.3 HIBERNATE Mode

The purpose of hibernate mode is to save the power. When in hibernate mode, CORE will no power consumption.

If you want to use hibernate mode, please first make sure RTCCR.SELEXC is 0.
When Software writes 1 to PD bit of HCR, the system at once enters HIBERNATE mode. The powers of CORE and IO are disconnected by PWRON pin, no power consumption in core and IO. When a wakeup event occurs, the core enters normal mode through a hibernate reset.

17.6.3.1 Procedure to Enter HIBERNATE mode

Before enter HIBERNATE mode, software must complete following steps:

1 Finish the current operation and preserve all data to flash.
2 Configure the wake-up sources properly by configure HWCR.
3 Set HIBERNATE MODE. (Set PD bit in HCR to 1)

17.6.3.2 Procedure to Wake-up from HIBERNATE mode

1 The internal hibernate reset signal will be asserted if one of the wake-up sources is issued.
2 Check RSR to determine what caused the reset.
3 Check PIN/ALM bits of HWRSR in order to know whether or not the power-up is caused by which wake-up from HIBERNATE mode.

4 Configure the SDRAM memory controller.
5 Recover the data from flash.

17.6.4 Time Regulation

Because of the inherent inaccuracy of crystal and other variables, the time counter may be inaccurate. This requires a slight adjustment. The application processor, through the RTCGR, lets you adjust the 1Hz time base to an error of less than 1ppm. Such that if the Hz clock were set to be 1Hz, there would be an error of less than 5 seconds per month.

To determine the value programmed into the RTCGR, you must first measure the output frequency at the oscillator multiplex (approximately 32 kHz) using an accurate time base, such as a frequency counter. This clock is externally visible by selecting the alternate function of GPIO.
To gain access to the clock, program this pin as an output and then switch to the alternate function.
To trim the clock, divide the output of the oscillator by an integer value and fractional adjust it by periodically deleting clocks from the stream driving this integer divider.

After the true frequency of the oscillator is known, it must be split into integer and fractional portions. The integer portion of the value (minus one) is loaded into the NC1HZ field of the RTCGR.

The fractional part of the adjustment is done by periodically deleting clocks from the clock stream driving the Hz divider. The trim interval period is hardwired to be 1024 1Hz clock cycles (approximately 17 minutes). The number of clocks (represented by ADJC field of RTCGR) are deleted from the input clock stream per trim interval. If ADJC is programmed to be zero, then no trim operations occur and the RTC is clocked with the raw 32 kHz clock. The relationship between the Hz clock frequency and the nominal 32 kHz clock (f1 and f32K, respectively) is shown in the following equation.

$$f1 = \frac{2^{10} \times (NC1HZ + 1)}{2^{10} \times (NC1HZ + 1) + ADJC} \times \frac{f32k}{NC1HZ + 1}$$

f1 = actual frequency of 1Hz clock
f32k = frequency of either 32.768KHz crystal output or 3.6864MHz crystal output further divided down to 32.914KHz

17.6.5 Clock select

There could be two clock input to RTC internal clock called rtclk. One is OSC32k clock; the other is EXCLK/512.
The software MUST make sure the RTC run in valid clock configuration.

Table 17-4 Clock select registers

RTCCR.SELEXC	CPM.ERCS	Description	Valid
0	0	RTC use OSC32K clock.	OK
0	1		OK
1	0	RTC use EXCLK/512 clock.	OK
1	1	RTC will lost clock. (Not Valid)	NO

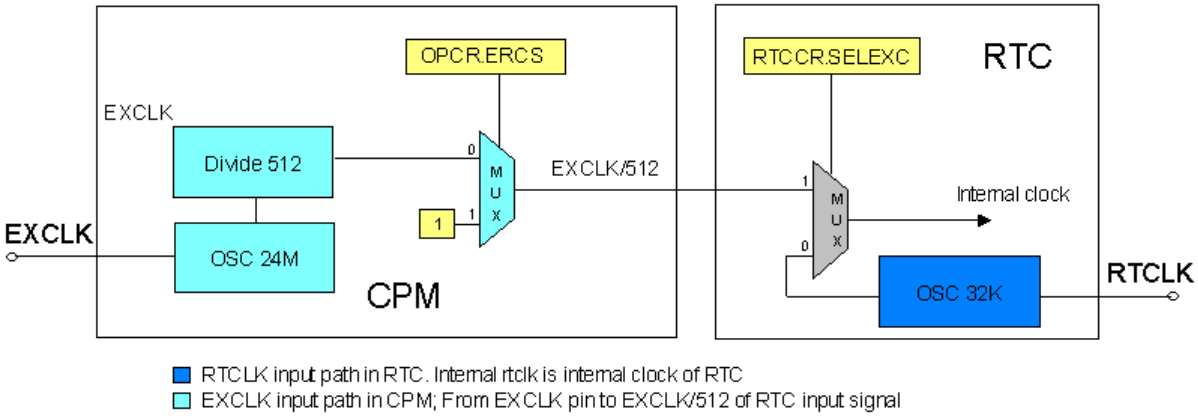


Figure 17-2 RTC clock selection path

Changing RTCLK sequence:

1 There are both 32KHz crystal and 24Mhz EXCLK crystal connected, so RTCLK input path has 32Khz clock.

In this case, there is no need to change internal clock, so do NOT change SELEXC all the time.

2 There is no 32KHz crystal connected but only 24Mhz EXCLK crystal connected, so RTCLK input path has no clock.

In this case, should flow the sequence below to change internal clock:

a Set OPCR.ERCS of CPM to 1; close EXCLK/512 to RTC.

b Set CLKGR.RTC of CPM to 0; open PCLK to RTC.

c Set RTCCR.SELEXC to 1; change internal clock to EXCLK/512.

d Wait two clock period of clock.

e Clear OPCR.ERCS of CPM to 0; open EXCLK/512 to RTC.

f Configure all RTC registers but RTCCR.SELEXC.

g Check RTCCR.SELEXC == 1.

h IF YES, finish this sequence; IF NO, do step (a) again.

NOTE: If using HIBERNATE mode, MUST have both 32KHz crystal (or input 32Khz clock) and 24Mhz EXCLK crystal connected, or RTC time will be insignificant.

18 EFUSE Slave Interface (EFUSE)

18.1 Overview

The EFUSE is provided 1K programmable bit, Total 1K bits are separated into eighteen segments, as below table show.

128bit	128bit	240bit	16bit	128bit	128bit	128bit	128bit

Each segment can be programmed separately.
Each segment has a protect bit, which has higher priority than program segment selection.

Programming frequency is wide, and programming time can adjust use the EFUSE register.

Initial value of EFUSE is 0, when programmed to 1, it doesn't able to program back to 0 anymore. Do not attempt to program any bit that already programmed to 1, such action will result unpredictable status to whole EFUSE block.

Programming voltage supply pin VDDQ:

- In program mode, supply VDDQ with 2.5V.
  Maximum accumulative time for VDDQ pin exposed under 2.5V+/-10% should be less than 1 sec.
- In read mode, leave VDDQ to 0V or floating.

18.2 Registers

Following chapter will descript the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute
   R - Read only
   W - Write only
   RW - read and write
   RCW - read and write, but clear to 0 by read
   RSW - read and write, but set to 1 by read
   RWC - read and write, clear to 0 by write 1, write 0 has no effect
   RWS - read and write, set to 1 by write 1, write 0 has no effect
   RC - read only, and clear to 0 by read
   RS - read only, and set to 1 by read

SPEC - special access method, relate to its description

3. Reset Value

1 - reset to 1

0 - reset to 0

? - value unknown after reset

18.2.1 Registers Memory Map

Table 18-1 Registers Memory Map-Address Base

Name	Base	Description
EFUSE	0x13540000	Address base of EFUSE

Table 18-2 EFUSE Register Description

Name	Description	Reset Value	Address Offset	Access Size
EFUCTRL	EFUSE Control Register	0x001F0000	0x0000	32
EFUCFG	EFUSE Configure Register	0x00000000	0x0004	32
EFUSTATE	EFUSE Status Register	0x00????00	0x0008	32
EFUDATA0	EFUSE Data 0 Register	0x0000????	0x000C	32
EFUDATA1	EFUSE Data 1 Register	0x00000000	0x0010	32
EFUDATA2	EFUSE Data 2 Register	0x00000000	0x0014	32
EFUDATA3	EFUSE Data 3 Register	0x00000000	0x0018	32
EFUDATA4	EFUSE Data 4 Register	0x00000000	0x001C	32
EFUDATA5	EFUSE Data 5 Register	0x00000000	0x0020	32
EFUDATA6	EFUSE Data 6 Register	0x00000000	0x0024	32
EFUDATA7	EFUSE Data 7 Register	0x00000000	0x0028	32

18.2.2 Registers and Fields Description

18.2.2.1 EFUSE Control Register (EFUCTRL)

	EFUCTRL																BASE + 0x0000															
Bit	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
	Reserved				address							Length					PG_EN	Reserved														WR_EN	RD_EN
RST	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:30	Reserved	Writing has no effect, read as zero.	R
27:21	Address	Indicate the start address wants to program or read Aligned to Byte	RW
20:16	Length	Indicate the length wants to program or read, efuse operate length equals to this value add 1. Byte number(min: 1byte, max: 32byte, default: 32byte)	RW
15	PG_EN	Programming EFUSE enable 1: Enable 0: Disable	RW
14:2	Reserved	Writing has no effect, read as zero.	RW
1	WR_EN	Write data to EFUSE data buffer enable 1: Enable 0: Disable This bit will change to 0 automatically when write procedure done	RW
0	RD_EN	Read data to EFUSE data buffer enable 1: Enable 0: Disable This bit will change to 0 automatically when read procedure done	RW

18.2.2.2 EFUSE Configure Register (EFUCFG)

	EFUCFG																BASE + 0x0004															
Bit	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
	INT_EN	Reserved									RD_ADJ		Reserved	RD_STROBE			Reserved		WR_ADJ		Reserved			WR_STROBE								
RST	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

Bits	Name	Description	RW
31	INT_EN	Interrupt enable when read/write EFUSE done. 1: Enable 0: Disable	RW
30:22	Reserved	Writing has no effect, read as zero.	R
21:20	RD_ADJ	Adjust number of h2clk cycles when EFUSE reading unsigned value, h2clk period multiply (RD_ADJ + 1) should greater than 2 ns.	RW
18:16	RD_STROBE	Adjust number of h2clk cycles when EFUSE reading STROBE Signed value, h2clk period multiply(RD_ADJ + 3 + RD_STROBE) should greater than 15 ns.	RW

13:12	WR_ADJ	Adjust number of h2clk cycles when EFUSE programming unsigned value, h2clk period multiply (WR_ADJ + 1) should greater than 2 ns.	RW
8:0	WR_STROBE	Adjust number of h2clk cycles when EFUSE programming STROBE Signed value, h2clk period multiply(WR_ADJ + 916 + WR_STROBE) should between 4us and 6us, typical 5us	RW

18.2.2.3 EFUSE Status Register (EFUSTATE)

	EFUSTATE																BASE + 0x0008													
Bit	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
	Reserved								UK_PRT	NKU_PRT	EXEY_EN	Reserved					CUT_PRT	CHID_PRT	Reserved	SEC_PRT	DIS_JTAG	Reserved		SCBT_EN	Reserved						WR_DONE	RD_DONE
RST	0	0	0	0	0	0	0	0	?	?	?	0	0	0	0	0	?	?	0	?	?	0	0	?	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
30:24	Reserved	Writing has no effect, read as zero.	R
23	UK_PRT	User key segment write protect 1: protect, this segment can't write 0: no protect, this segment can write	R
22	NKU_PRT	NKU segment write protect 1: protect, this segment can't write 0: no protect, this segment can write	R
21	EXKEY_EN	External key enable 1: The security boot use external key 0: The security boot use internal key	R
20:16	Reserved	Writing has no effect, read as zero.	R
15	CUSTID_PRT	Customer ID segment write protect bit 1: protect, this segment can't write 0: no protect, this segment can write	R
14	CHIPID_PRT	CHIP ID segment protect bit 1: protect, this segment can't write 0: no protect, this segment can write	R
13	Reserved	Writing has no effect, read as zero.	R
12	SECBOOT_PRT	Security boot enable protect bit 1: protect, security boot enable can write 0: no protect, security boot enable can't write	R
11	DIS_JTAG	Disable JTAG 1: disable JTAG 0: enable JTAG	R

10:9	Reserved	Writing has no effect, read as zero.	RW
8	SECBOOT_EN	Security boot enable 1: enable Security boot 0: Normal boot	R
7:2	Reserved	Writing has no effect, read as zero.	RW
1	WR_DONE	Write data to EFUSE operate done flag 1: done 0: not done *This bit only can write to 0, and write to 0 will clear this flag.	RW
0	RD_DONE	Read EFUSE data operate done flag 1: done 0: not done *This bit only can write to 0, and write to 0 will clear this flag	RW

18.2.2.4 EFUSE Data Register (EFUDATAn)

	EFUDATAn (n=0,1,2,3,4,5,6,7)																BASE + 0x000C ~ BASE + 0x0028														
Bit	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
	DATA																															
RST	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

Bits	Name	Description	RW
31:0	DATA	EFUSE DATA register. *EFUSE DATA0 register mapped low operate address, others mapped increased address.	RW

NOTE:

1. The EFUSE 1K bits are mapped to chip memory address, from 0x13540200 to 0x1354027F.
2. The EFUSE 1K programmable bits are separated into thirteen segments as below table.

The first Segment used to store Ingenic chip id, second segment store 128bit random number, third segment used to store customer id, forth segment store segment protect bit, fifth segment used to store root key, sixth segment used to store chip key, seventh used to sore user key, the last segment used to store NKU.

Segment	1	2	3	4	5	6	7	8
Function	CHIP ID	RN	CUT ID	PRT	RK	CK	UK	NKU
Mapped	200	210	220	23E	240	250	260	270

Address	20F	21F	23D	23F	24F	25F	26F	27F
Bits	128	128	240	16	128	128	128	128

3. The protect bit segment store others segment write protect bit, if protect bit programmed to 1, corresponding segment can not be able to program. The protect bit map as below table

	Bit	Mapped To EFUSE Status Register
The Protect bit Segment	15	UK_PRT
	14	NKU_PRT
	13	EXKEY_EN
	12	Reserved
	11	Reserved
	10	Reserved
	9	Reserved
	8	Reserved
	7	USERID_PRT
	6	CHIPID_PRT
	5	Reserved
	4	SECBOOT_PRT
	3	DISJTAG
	2	Reserved
	1	Reserved
	0	SECBOOT_EN

4. The EFUSE can read or write maximum 256bit in each time, but we strongly recommend operate each segment separately.

5. The segment(Root key/Chip key/User key/NKU) can only read/write by SC-ROM.

18.3 Flow

18.3.1 Program EFUSE Flow

1. Set EFUCFG register(reading and programming timing).
2. Write want program data to EFUSE data buffer 0-7 registers
3. Set control register, indicate want to program address, data length.
4. Write control register PG_EN bit to 1
5. Connect VDDQ pin to 2.5V
6. Write control register WR_EN bit to 1
7. Wait status register WR_DONE set to 1.
8. Disconnect VDDQ pin from 2.5V.
9. Write control register PG_EN bit to 0.

18.3.2 Program Security Key Flow

1. Set EFUCFG register(reading and programming timing).
2. Write control register PG_EN bit to 1
3. Connect VDDQ pin to 2.5V
4. Invoke SC-ROM controller to write data to Security key mapped address
5. Wait status register WR_DONE set to 1
6. jump to step 4, program all data like this
7. Disconnect VDDQ pin from 2.5V.
8. Write control register PG_EN bit to 0.

NOTE:

*[1]: Do NOT pre-charge VDDQ too early before EFUSE program started.
The maximum 2.5V supply time to VDDQ must be strictly controlled less than 1sec.
Use on-chip counter and GPIO to coordinate external supply source.

*[2]: Only the 1.3.2 flow can be used to program the User Key and NKU Segment, The Root key and Chip key Segment will be programmed at CP by Ingenic, other segments only can program use 1.3.1 flow.

*[3]: The SC-ROM Controller can write Key segment 32bits once time.

18.3.3 Read EFUSE Flow

1. Set EFUCFG register(reading and programming timing).
2. Set control register to indicate what to read data address, read data length.
3. Set read enable register.
4. Wait status register RD_DONE set to 1 or EFUSE interrupted, software application can read EFUSE data buffer 0 – 7 registers.

18.3.4 Read Security Key/Random Number Flow

1. Invoke SC-ROM controller to read corresponding map address.

NOTE:

*[1]: EFSUE controller will read EFUSE protect bit segment automatically when chip reset, and update the EFUSE state register.

*[2]: Only 1.3.4 flow can be used to read all Keys, other segments only can read use 1.3.3 flow.

*[3]: The SC-ROM Controller can read Keys value 32bits once time.

Section 7

PERIPHERALS

19 General-Purpose I/O Ports

19.1 Overview

General Purpose I/O Ports (GPIO) is used in generating and capturing application-specific input and output signals. Each port can be programmed as an output, an input or function port that serves certain peripheral. As input, pull up/down can be enabled/disabled for the port and the port also can be configured as level or edge tripped interrupt source.

19.2 Features

- Each port can be configured as an input, an output or an alternate function port.
- Each port can be configured as an interrupt source of low/high level or rising/falling edge triggering, and every interrupt source can be masked independently.
- Each port has an internal pull-up or pull-down resistor connected. The pull-up/down resistor can be disabled.
- GPIO output 4 interrupts, each group had an interrupt to INTC(Interrupt Controller).

19.3 About GPIO Port Summary Table

- In the GPIO port summary, different pad type will determine different pad characteristics.
- Each Pad can configured as four module function.

Table 19-1 GPIO port summary table illustration

PAD_ID	PAD_TYPE	PAD_PULL_ACT	PULL_RST	ST_RST	SL_RST	DS_RST	FUNC0	FUNC1	FUNC2	FUNC3
PA00	PRUW08DGZ_G	0	1	-	-	-	sd0(io-0)	slcd_d0(o)	smb1_sck(io-1)	(i-0)
PB25	PRDW08DGZ_G	0	0	-	-	-	drv_vbus_o(o)	dmic2_in(i-0)	(i-0)	(i-0)
PC16	PRUW08SDGZ_G	0	1	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PC23	PRUW08SDGZ_G	0	Z	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PD00	PRUW08DGZ	0	1	-	-	-	ssi0_clk(o)	smb2_sck(io-1)	(i-0)	(i-0)
Pad name	Pad instance type name (different type will decide	Pad pull active value. 1: pull enable is high level	After reset, pad default pull enable value	After reset, Schmitt trigger enable default value	After reset, slew rate enable default value	After reset, driving strength default value	In table, each function column will expression one module port using relationship. such as the pad PA00, the FUNC0 "sd0_(io-0)", "sd0"			

	GPIO function)	active 0: pull enable is low level active	The parameter means of the above options are as follows: 1: reset value is high level voltage. 0: reset value is low level voltage. -: this feature does not exit. Z: do nothing.	means the emc module data port 0, "io" means function as I/O, "0" means when configure as other functions, GPIO supply value 0 to this function port.

NOTES:

1. Feature "PAD_PULL_ACT" in table just for RTL verification. software program should don't care about this information.
2. Feature "PULL_RST, ST_RST, SL_RST, DS_RST " means after reset default value.
3. The PAD_TYPE of "PRU* " is pull-up function pad, and the PAD_TYPE of "PRD* " is pull-down function pad.
4. The PAD_TYPE of "*SDGZ* " is Schmitt trigger input pad, ant the others type are normal pad.

19.3.1 GPIO Port A Summary

Table 19-2 GPIO Port A summary

#PAD_ID	PAD_TYPE	AD_PULL_ACT	PULL_RST	ST_RST	SL_RST	DS_RST	FUNCTION0	FUNCTION1	FUNCTION2	FUNCTION3
PA00	PRUW08DGZ_G	0	1	-	-	-	sd0(io-0)	slcd_d0(o)	smb1_sck(io-1)	(i-0)
PA01	PRUW08DGZ_G	0	1	-	-	-	sd1(io-0)	slcd_d1(o)	smb1_sda(io-1)	(i-0)
PA02	PRUW08DGZ_G	0	1	-	-	-	sd2(io-0)	slcd_d2(o)	uart2_rxd(i-1)	(i-0)
PA03	PRUW08DGZ_G	0	1	-	-	-	sd3(io-0)	slcd_d3(o)	uart2_txd(o)	(i-0)
PA04	PRUW08DGZ_G	0	1	-	-	-	sd4(io-0)	slcd_d4(o)	uart1_rxd(i-1)	(i-0)
PA05	PRUW08DGZ_G	0	1	-	-	-	sd5(io-0)	slcd_d5(o)	uart1_txd(o)	(i-0)
PA06	PRUW08DGZ_G	0	1	-	-	-	sd6(io-0)	slcd_d6(o)	(i-0)	(i-0)
PA07	PRUW08DGZ_G	0	1	-	-	-	sd7(io-0)	slcd_d7(o)	(i-0)	(i-0)
PA08	PRUW08DGZ_G	0	1	-	-	-	sd8(io-0)	slcd_d8(o)	cim_pclk_i(i-0)	(i-0)
PA09	PRUW08DGZ_G	0	1	-	-	-	sd9(io-0)	slcd_d9(o)	cim_hsyn_i(i-0)	(i-0)
PA10	PRUW08DGZ_G	0	1	-	-	-	sd10(io-0)	slcd_d10(o)	cim_vsyn_i(i-0)	(i-0)
PA11	PRUW08DGZ_G	0	1	-	-	-	sd11(io-0)	slcd_d11(o)	cim_mclk_o(o)	(i-0)
PA12	PRUW08DGZ_G	0	1	-	-	-	sd12(io-0)	slcd_d12(o)	cim_d7_i(i-0)	(i-0)
PA13	PRUW08DGZ_G	0	1	-	-	-	sd13(io-0)	slcd_d13(o)	cim_d6_i(i-0)	(i-0)
PA14	PRUW08DGZ_G	0	1	-	-	-	sd14(io-0)	slcd_d14(o)	cim_d5_i(i-0)	(i-0)
PA15	PRUW08DGZ_G	0	1	-	-	-	sd15(io-0)	slcd_d15(o)	cim_d4_i(i-0)	(i-0)
PA16	PRUW08DGZ_G	0	1	-	-	-	(i-0)	msc0_d7(io-1)	cim_d3_i(i-0)	(i-0)
PA17	PRUW08DGZ_G	0	1	-	-	-	(i-0)	msc0_d6(io-1)	cim_d2_i(i-0)	(i-0)
PA18	PRUW08DGZ_G	0	1	-	-	-	(i-0)	msc0_d5(io-1)	cim_d1_i(i-0)	(i-0)
PA19	PRUW08DGZ_G	0	1	-	-	-	(i-0)	msc0_d4(io-1)	cim_d0_i(i-0)	(i-0)
PA20	PRUW08DGZ_G	0	1	-	-	-	(i-0)	msc0_d3(io-1)	ssi0_gpc(o)	(i-0)
PA21	PRUW08DGZ_G	0	1	-	-	-	(i-0)	msc0_d2(io-1)	ssi0_ce1(o)	(i-0)
PA22	PRUW08DGZ_G	0	1	-	-	-	(i-0)	msc0_d1(io-1)	ssi0_dt(o)	(i-0)
PA23	PRUW08DGZ_G	0	1	-	-	-	(i-0)	msc0_d0(io-1)	ssi0_dr(i-0)	(i-0)
PA24	PRUW08DGZ_G	0	1	-	-	-	(i-0)	msc0_clk(o)	ssi0_clk(o)	(i-0)
PA25	PRUW08DGZ_G	0	1	-	-	-	(i-0)	msc0_cmd(io-1)	ssi0_ce0(o)	(i-0)
PA26	PRUW08DGZ_G	0	1	-	-	-	(i-0)	sfc_clk(o)	ssi0_clk(o)	(i-0)
PA27	PRUW08DGZ_G	0	1	-	-	-	(i-0)	sfc_ce(o)	ssi0_ce0(o)	(i-0)
PA28	PRUW08DGZ_G	0	1	-	-	-	(i-0)	sfc_dr(io-0)	ssi0_dr(i-0)	(i-0)
PA29	PRUW08DGZ_G	0	1	-	-	-	(i-0)	sfc_dt(io-0)	ssi0_dt(o)	(i-0)
PA30	PRUW08DGZ_G	0	1	-	-	-	(i-0)	sfc_wp(io-0)	ssi0_ce1(o)	(i-0)
PA31	PRUW08DGZ_G	0	1	-	-	-	(i-0)	sfc_hold(io-0)	ssi0_gpc(o)	(i-0)

19.3.2 GPIO Port B Summary

Table 19-3 GPIO Port B summary

#PAD_ID	PAD_TYPE	AD_PULL_AC	PULL_RST	ST_RST	SL_RST	DS_RST	FUNCTION0	FUNCTION1	FUNCTION2	FUNCTION3
PB00	PRUW08DGZ_G	0	1	-	-	-	sa0_o(o)	i2s_mclk(o)	(i-0)	(i-0)
PB01	PRUW08DGZ_G	0	1	-	-	-	sa1_o(o)	i2s_bclk(io-0)	(i-0)	(i-0)
PB02	PRUW08DGZ_G	0	1	-	-	-	sa2_o(o)	i2s_lrclk(io-0)	(i-0)	(i-0)
PB03	PRUW08DGZ_G	0	1	-	-	-	sa3_o(o)	i2s_di(i-0)	(i-0)	(i-0)
PB04	PRUW08DGZ_G	0	1	-	-	-	sa4_o(o)	i2s_do(o)	(i-0)	(i-0)
PB05	PRUW08DGZ_G	0	1	-	-	-	sa5_o(o)	dmic1_in(i-0)	(i-0)	(i-0)
PB06	PRUW08DGZ_G	0	1	-	-	-	sa6_o(o)	mac_phy_clk(o)	pwm3(io-0)	(i-0)
PB07	PRUW08DGZ_G	0	1	-	-	-	sa7_o(o)	mac_crs_dv(i-0)	(i-0)	(i-0)
PB08	PRUW08DGZ_G	0	1	-	-	-	sa8_o(o)	mac_rxd1(i-0)	(i-0)	(i-0)
PB09	PRUW08DGZ_G	0	1	-	-	-	sa9_o(o)	mac_rxd0(i-0)	(i-0)	(i-0)
PB10	PRUW08DGZ_G	0	1	-	-	-	sa10_o(o)	mac_txen(o)	(i-0)	(i-0)
PB11	PRUW08DGZ_G	0	1	-	-	-	sa11_o(o)	mac_txd1(o)	(i-0)	(i-0)
PB12	PRUW08DGZ_G	0	1	-	-	-	sa12_o(o)	mac_txd0(o)	(i-0)	(i-0)
PB13	PRUW08DGZ_G	0	1	-	-	-	sa13_o(o)	mac_mdc(o)	(i-0)	(i-0)
PB14	PRUW08DGZ_G	0	1	-	-	-	sa14_o(o)	mac_mdio(io-0)	(i-0)	(i-0)
PB15	PRUW08DGZ_G	0	1	-	-	-	sa15_o(o)	mac_ref_clk(i-0)	(i-0)	(i-0)
PB16	PRUW08DGZ_G	0	1	-	-	-	rd_o(o)	slcd_rd(o)	(i-0)	(i-0)
PB17	PRUW08DGZ_G	0	1	-	-	-	we_o(o)	slcd_wr(o)	(i-0)	(i-0)
PB18	PRUW08DGZ_G	0	1	-	-	-	cs1_o(o)	slcd_ce(o)	(i-0)	(i-0)
PB19	PRUW08DGZ_G	0	1	-	-	-	cs2_o(o)	slcd_te(i-1)	(i-0)	(i-0)
PB20	PRUW08DGZ_G	0	1	-	-	-	wait_i(i-1)	slcd_dc(o)	(i-0)	(i-0)
PB21	PRUW08DGZ_G	0	1	-	-	-	dmic_clk_o(o)	(i-0)	(i-0)	(i-0)
PB22	PRUW08DGZ_G	0	1	-	-	-	dmic0_in(i-0)	(i-0)	(i-0)	(i-0)
PB23	PRUW08DGZ_G	0	1	-	-	-	smb0_sck(io-1)	scc_clk(o)	(i-0)	(i-0)
PB24	PRUW08DGZ_G	0	1	-	-	-	smb0_sda(io-1)	scc_data(io-1)	(i-0)	(i-0)
PB25	PRDW08DGZ_G	0	0	-	-	-	drv_vbus_o(o)	(i-0)	(i-0)	(i-0)
PB26	PRUW08DGZ_G	0	1	-	-	-	rtc32k(o)	(i-0)	(i-0)	(i-0)
PB27	PRUW08DGZ_G	0	1	-	-	-	exclk(o)	(i-0)	(i-0)	(i-0)
PB28	PRUW08DGZ_G	0	Z	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PB29	PRUW08DGZ_G	0	Z	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PB30	PRUW08DGZ_G	0	Z	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PB31	PRUW08SDGZ_G	0	1	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)

NOTES:

1 PB28: GPIO group B bit 28 is used as BOOT_SEL0 input during boot.

2 PB29: GPIO group B bit 29 is used as BOOT_SEL1 input during boot.

3 PB30: GPIO group B bit 30 is used as BOOT_SEL2 input during boot.

4 BOOT_SEL2, BOOT_SEL1, BOOT_SEL0 are used to select boot source and function during the processor boot.

5 PB28、PB29、PB30：after system startup, it can used as normal GPIO function.

6 PB31 is RTC WKUP pin, can only be used as input and interrupt, no pull-up and pull-down.

7 The SLCD rd and ce function only can be used by set PB16/PB18 as normal GPIO function.

19.3.3 GPIO Port C Summary

Table 19-4 GPIO Port C summary

#PAD_ID	PAD_TYPE	AD_PULL_AC	PULL_RST	ST_RST	SL_RST	DS_RST	FUNCTION0	FUNCTION1	FUNCTION2	FUNCTION3
PC00	PRUW08DGZ_G	0	1	-	-	-	msc1_clk(o)	(i-0)	(i-0)	(i-0)
PC01	PRUW08DGZ_G	0	1	-	-	-	msc1_cmd(io-1)	(i-0)	(i-0)	(i-0)
PC02	PRUW08DGZ_G	0	1	-	-	-	msc1_d0(io-1)	(i-0)	(i-0)	(i-0)
PC03	PRUW08DGZ_G	0	1	-	-	-	msc1_d1(io-1)	(i-0)	(i-0)	(i-0)
PC04	PRUW08DGZ_G	0	1	-	-	-	msc1_d2(io-1)	(i-0)	(i-0)	(i-0)
PC05	PRUW08DGZ_G	0	1	-	-	-	msc1_d3(io-1)	(i-0)	(i-0)	(i-0)
PC06	PRUW08DGZ_G	0	1	-	-	-	pcm_clk(io-0)	(i-0)	(i-0)	(i-0)
PC07	PRUW08DGZ_G	0	1	-	-	-	pcm_do(o)	(i-0)	(i-0)	(i-0)
PC08	PRUW08DGZ_G	0	1	-	-	-	pcm_di(i-0)	(i-0)	(i-0)	(i-0)
PC09	PRUW08DGZ_G	0	1	-	-	-	pcm_syn(io-0)	(i-0)	(i-0)	(i-0)
PC10	PRUW08DGZ_G	0	1	-	-	-	uart0_rxd(i-1)	(i-0)	(i-0)	(i-0)
PC11	PRUW08DGZ_G	0	1	-	-	-	uart0_txd(o)	(i-0)	(i-0)	(i-0)
PC12	PRUW08DGZ_G	0	1	-	-	-	uart0_cts(i-0)	(i-0)	(i-0)	(i-0)
PC13	PRUW08DGZ_G	0	1	-	-	-	uart0_rts(o)	(i-0)	(i-0)	(i-0)
PC16	PRUW08SDGZ_G	0	1	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PC17	PRUW08SDGZ_G	0	1	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PC18	PRUW08SDGZ_G	0	1	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PC19	PRUW08SDGZ_G	0	1	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PC20	PRUW08SDGZ_G	0	1	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PC21	PRUW08SDGZ_G	0	1	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PC22	PRUW08SDGZ_G	0	Z	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PC23	PRUW08SDGZ_G	0	Z	-	-	-	(i-0)	(i-0)	(i-0)	(i-0)
PC24	PRUW08DGZ_G	0	1	-	-	-	pwm4(io-0)	(i-0)	(i-0)	(i-0)
PC25	PRDW08DGZ_G	0	0	-	-	-	pwm0(io-0)	(i-0)	(i-0)	(i-0)
PC26	PRUW08DGZ_G	0	1	-	-	-	smb1_sck(io-1)	pwm1(o)	(i-0)	(i-0)
PC27	PRUW08DGZ_G	0	1	-	-	-	smb1_sda(io-1)	pwm2(o)	(i-0)	(i-0)

NOTES:

1. PC31 is only used to select the function between UART and JTAG of JTAG/UART2 Pins(TDI_UART2_RXD and TDO_UART2_TXD), No corresponding pin exists for this GPIO.
   When PC31.function1 is false, select JTAG function.
   When PC31.function1 is true, select UART2 function.
2. The pull enable of PC31 is used to control UART and JTAG of JTAG/UART2 Pins.
3. The PC16-PC23 implement glitch filter for interrupt input mode.

19.3.4 GPIO Port D Summary

Table 19-5 GPIO Port D summary

#PAD_ID	PAD_TYPE	AD_PULL_AC	PULL_RST	ST_RST	SL_RST	DS_RST	FUNCTION0	FUNCTION1	FUNCTION2	FUNCTION3
PD00	PRUW08DGZ	0	1	-	-	-	ssi0_clk(o)	smb2_sck(io-1)	(i-0)	(i-0)
PD01	PRUW08DGZ	0	1	-	-	-	ssi0_ce0(o)	smb2_sda(io-1)	(i-0)	(i-0)
PD02	PRUW08DGZ	0	1	-	-	-	ssi0_dt(o)	uart1_rxd(i-1)	(i-0)	(i-0)
PD03	PRUW08DGZ	0	1	-	-	-	ssi0_dr(i-0)	uart1_txd(o)	(i-0)	(i-0)
PD04	PRUW08DGZ	0	1	-	-	-	uart2_txd(o)	uart1_cts(i-0)	(i-0)	(i-0)
PD05	PRUW08DGZ	0	1	-	-	-	uart2_rxd(i-1)	uart1_rts(o)	(i-0)	(i-0)

NOTES: All group D pins are 5V-Tolerant, the max input high voltage is 5.5V.

19.3.5 GPIO Port Z - Shadow Group

The group is a dummy group which is not mapping to any on-chip ports. It is used to avoid 3rd (intermittent) state error during configuration. Please refer 1.4.5 and 1.4.6 for detailed information.

19.4 Registers Description

Following chapter will descript the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute
   R - Read only
   W - Write only
   RW - read and write
   RCW - read and write, but clear to 0 by read
   RSW - read and write, but set to 1 by read
   RWC - read and write, clear to 0 by write 1, write 0 has no effect
   RWS - read and write, set to 1 by write 1, write 0 has no effect
   RC - read only, and clear to 0 by read
   RS - read only, and set to 1 by read
   SPEC - special access method, relate to its description
3. Reset Value
   1 - reset to 1
   0 - reset to 0
   ? - value unknown after reset
4. Not all the register will be use in this chapter, please review the chapter of "Function Overview" for more detail.

Table 19-6 summarized all memory-mapped registers, which can be programmed to operate GPIO port and alternate function port sharing configuration.

All registers are in 32-bits width. Usually, 1 bit in the register affects a corresponding GPIO port and every GPIO port can be operated independently.

Gray bits in the register description indicates that this feature in the GPIO group who did not use, program can be ignored.

Table 19-6 Registers Memory Map-Address Base

Name	Base	Description
PA	0x10010000	Address base of GPIO Port A
PB	0x10010100	Address base of GPIO Port B
PC	0x10010200	Address base of GPIO Port C
PD	0x10010300	Address base of GPIO Port D
PZ	0x10010700	Address base of GPIO Group Z (Shadow group)

Table 19-7 GPIO Registers

Name	RW	Description	Reset Value	Address Offset	Access Size
GPIO PORT A					
PAPIN	R	PORT A PIN Level Register	0x00000000	0x00	32
PAINT	RW	PORT A Interrupt Register	0x00000000	0x10	32
PAINTS	W	PORT A Interrupt Set Register	0x00000000	0x14	32
PAINTC	W	PORT A Interrupt Clear Register	0x00000000	0x18	32
PAMSK	RW	PORT A Interrupt Mask Register	0xFFFFFFFF	0x20	32
PAMSKS	W	PORT A Interrupt Mask Set Register	0x00000000	0x24	32
PAMSKC	W	PORT A Interrupt Mask Clear Register	0x00000000	0x28	32
PAPAT1	RW	PORT A Pattern 1 Register	0xFFFFFFFF	0x30	32
PAPAT1S	W	PORT A Pattern 1 Set Register	0x00000000	0x34	32
PAPAT1C	W	PORT A Pattern 1 Clear Register	0x00000000	0x38	32
PAPAT0	RW	PORT A Pattern 0 Register	0x00000000	0x40	32
PAPAT0S	W	PORT A Pattern 0 Set Register	0x00000000	0x44	32
PAPAT0C	W	PORT A Pattern 0 Clear Register	0x00000000	0x48	32
PAFLG	R	PORT A FLAG Register	0x00000000	0x50	32
PAFLGC	W	PORT A FLAG Clear Register	0x00000000	0x58	32
PAPEN	RW	PORT A PULL Disable Register	0x00000000	0x70	32
PAPENS	W	PORT A PULL Disable Set Register	0x00000000	0x74	32
PAPENC	W	PORT A PULL Disable Clear Register	0x00000000	0x78	32
GPIO PORT B					
PBPIN	R	PORT B PIN Level Register	0x00000000	0x00	32
PBINT	RW	PORT B Interrupt Register	0x00000000	0x10	32
PBINTS	W	PORT B Interrupt Set Register	0x00000000	0x14	32
PBINTC	W	PORT B Interrupt Clear Register	0x00000000	0x18	32
PBMSK	RW	PORT B Interrupt Mask Register	0xFFFFFFFF	0x20	32
PBMSKS	W	PORT B Interrupt Mask Set Register	0x00000000	0x24	32
PBMSKC	W	PORT B Interrupt Mask Clear Register	0x00000000	0x28	32
PBPAT1	RW	PORT B Pattern 1 Register	0xFFFFFFFF	0x30	32
PBPAT1S	W	PORT B Pattern 1 Set Register	0x00000000	0x34	32
PBPAT1C	W	PORT B Pattern 1 Clear Register	0x00000000	0x38	32
PBPAT0	RW	PORT B Pattern 0 Register	0x00000000	0x40	32
PBPAT0S	W	PORT B Pattern 0 Set Register	0x00000000	0x44	32
PBPAT0C	W	PORT B Pattern 0 Clear Register	0x00000000	0x48	32
PBFLG	R	PORT B FLAG Register	0x00000000	0x50	32
PBFLGC	W	PORT B FLAG Clear Register	0x00000000	0x58	32
PBPEN	RW	PORT B PULL Disable Register	0x70000000	0x70	32
PBPENS	W	PORT B PULL Disable Set Register	0x00000000	0x74	32
PBPENC	W	PORT B PULL Disable Clear Register	0x00000000	0x78	32

GPIO PORT C					
PCPIN	R	PORT C PIN Level Register	0x00000000	0x00	32
PCINT	RW	PORT C Interrupt Register	0x00000000	0x10	32
PCINTS	W	PORT C Interrupt Set Register	0x00000000	0x14	32
PCINTC	W	PORT C Interrupt Clear Register	0x00000000	0x18	32
PCMSK	RW	PORT C Interrupt Mask Register	0x8FFF3FFF	0x20	32
PCMSKS	W	PORT C Interrupt Mask Set Register	0x00000000	0x24	32
PCMSKC	W	PORT C Interrupt Mask Clear Register	0x00000000	0x28	32
PCPAT1	RW	PORT C Pattern 1 Register	0x8FFF3FFF	0x30	32
PCPAT1S	W	PORT C Pattern 1 Set Register	0x00000000	0x34	32
PCPAT1C	W	PORT C Pattern 1 Clear Register	0x00000000	0x38	32
PCPAT0	RW	PORT C Pattern 0 Register	0x00000000	0x40	32
PCPAT0S	W	PORT C Pattern 0 Set Register	0x00000000	0x44	32
PCPAT0C	W	PORT C Pattern 0 Clear Register	0x00000000	0x48	32
PCFLG	R	PORT C FLAG Register	0x00000000	0x50	32
PCFLGC	W	PORT C FLAG Clear Register	0x00000000	0x58	32
PCPEN	RW	PORT C PULL Disable Register	0x00C00000	0x70	32
PCPENS	W	PORT C PULL Disable Set Register	0x00000000	0x74	32
PCPENC	W	PORT C PULL Disable Clear Register	0x00000000	0x78	32
PCGFCFG0	RW	PORT C Glitch Filter Configure 0	0x00000000	0x800	32
PCGFCFG0S	W	PORT C Glitch Filter Configure 0 Set	0x00000000	0x804	32
PCGFCFG0C	W	PORT C Glitch Filter Configure 0 Clear	0x00000000	0x808	32
PCGFCFG1	RW	PORT C Glitch Filter Configure 1	0x00000000	0x810	32
PCGFCFG1S	W	PORT C Glitch Filter Configure 1 Set	0x00000000	0x814	32
PCGFCFG1C	W	PORT C Glitch Filter Configure 1 Clear	0x00000000	0x818	32
PCGFCFG2	RW	PORT C Glitch Filter Configure 2	0x00000000	0x820	32
PCGFCFG2S	W	PORT C Glitch Filter Configure 2 Set	0x00000000	0x824	32
PCGFCFG2C	W	PORT C Glitch Filter Configure 2 Clear	0x00000000	0x828	32
PCGFCFG3	RW	PORT C Glitch Filter Configure 3	0x00000000	0x830	32
PCGFCFG3S	W	PORT C Glitch Filter Configure 3 Set	0x00000000	0x834	32
PCGFCFG3C	W	PORT C Glitch Filter Configure 3 Clear	0x00000000	0x838	32
GPIO PORT D					
PDPIN	R	PORT D PIN Level Register	0x00000000	0x00	32
PDINT	RW	PORT D Interrupt Register	0x00000000	0x10	32
PDINTS	W	PORT D Interrupt Set Register	0x00000000	0x14	32
PDINTC	W	PORT D Interrupt Clear Register	0x00000000	0x18	32
PDMSK	RW	PORT D Interrupt Mask Register	0x0000003F	0x20	32
PDMSKS	W	PORT D Interrupt Mask Set Register	0x00000000	0x24	32
PDMSKC	W	PORT D Interrupt Mask Clear Register	0x00000000	0x28	32
PDPAT1	RW	PORT D Pattern 1 Register	0x0000003F	0x30	32
PDPAT1S	W	PORT D Pattern 1 Set Register	0x00000000	0x34	32
PDPAT1C	W	PORT D Pattern 1 Clear Register	0x00000000	0x38	32

PDPAT0	RW	PORT D Pattern 0 Register	0x00000000	0x40	32
PDPAT0S	W	PORT D Pattern 0 Set Register	0x00000000	0x44	32
PDPAT0C	W	PORT D Pattern 0 Clear Register	0x00000000	0x48	32
PDFLG	R	PORT D FLAG Register	0x00000000	0x50	32
PDFLGC	W	PORT D FLAG Clear Register	0x00000000	0x58	32
PDPEN	RW	PORT D PULL Disable Register	0x00000000	0x70	32
PDPENS	W	PORT D PULL Disable Set Register	0x00000000	0x74	32
PDPENC	W	PORT D PULL Disable Clear Register	0x00000000	0x78	32
Shadow Register Group PZ					
PzINTS	W	Shadow Interrupt Set Register	0x00000000	0x14	32
PzINTC	W	Shadow Interrupt Clear Register	0x00000000	0x18	32
PzMSKS	W	Shadow Interrupt Mask Set Register	0x00000000	0x24	32
PzMSKC	W	Shadow Interrupt Mask Clear Register	0x00000000	0x28	32
PzPAT1S	W	Shadow Interrupt Pattern 1 Set Register	0x00000000	0x34	32
PzPAT1C	W	Shadow Interrupt Pattern 1 Clear Register	0x00000000	0x38	32
PzPAT0S	W	Shadow Interrupt Pattern 0 Set Register	0x00000000	0x44	32
PzPAT0C	W	Shadow Interrupt Pattern 0 Clear Register	0x00000000	0x48	32
PzGID2LD	W	PORT Group ID to be load from PZ Register	0x00000000	0xF0	32

NOTE:

1. Pz**** in the description of register is use for shadow operation for other groups, more details will description as follow chapters.

19.4.1 PORT A Register Group

19.4.1.1 PORT A PIN Level Registers (PAPINL,0x00)

PORT A GROUP level registers. They are read-only registers.

	PAPINL																														0x00
Bit	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
	PINL31	PINL30	PINL29	PINL28	PINL27	PINL26	PINL25	PINL24	PINL23	PINL22	PINL21	PINL20	PINL19	PINL18	PINL17	PINL16	PINL15	PINL14	PINL13	PINL12	PINL11	PINL10	PINL09	PINL08	PINL07	PINL06	PINL05	PINL04	PINL03	PINL02	PINL01	PINL00
RST	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

Bits	Name	Description	R/W
31:0	PINL n	Where n = 0 ~ 31 and PAPINL n = PAPINL0 ~ PAPINL31. The PORT A PIN level can be read by reading PINL n bit in register PAPINL.	R

19.4.1.2 PORT A Interrupt Registers (PAINT,0x10)

PORT A GROUP interrupt registers.

	PAINT																													0x10
Bit	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
	INT31	INT30	INT29	INT28	INT27	INT26	INT25	INT24	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16	INT15	INT14	INT13	INT12	INT11	INT10	INT09	INT08	INT07	INT06	INT05	INT04	INT03	INT02	INT01	INT00
RST	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

Bits	Name	Description	R/W
31:0	INT n	Where n = 0 ~ 31 and INT n = INT31 ~ INT00. Interrupt enable. 0: Corresponding pin is used as device functions or normal gpio 1: Corresponding pin is used as interrupt	RW

19.4.1.3 PORT A Interrupt Set Registers (PAINTS,0x14)

PORT A GROUP interrupt set registers.

	PAINTS																													0x14
Bit	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
	INTS31	INTS30	INTS29	INTS28	INTS27	INTS26	INTS25	INTS24	INTS23	INTS22	INTS21	INTS20	INTS19	INTS18	INTS17	INTS16	INTS15	INTS14	INTS13	INTS12	INTS11	INTS10	INTS09	INTS08	INTS07	INTS06	INTS05	INTS04	INTS03	INTS02	INTS01	INTS00
RST	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

Bits	Name	Description	R/W
31:0	INTS n	Writing 1 to INTS n will set INT n to 1 in register PAINT. Writing 0 to INTS n will no use.	WC

19.4.1.4 PORT A Interrupt Clear Registers (PAINTC,0x18)

PORT A GROUP interrupt clear registers.

	PAINTC																													0x18
Bit	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
	INTC31	INTC30	INTC29	INTC28	INTC27	INTC26	INTC25	INTC24	INTC23	INTC22	INTC21	INTC20	INTC19	INTC18	INTC17	INTC16	INTC15	INTC14	INTC13	INTC12	INTC11	INTC10	INTC09	INTC08	INTC07	INTC06	INTC05	INTC04	INTC03	INTC02	INTC01	INTC00
RST	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

Bits	Name	Description	R/W
31:0	INTC n	Writing 1 to INTC n will set INT n to 0 in register PAINT. Writing 0 to INTC n will no use.	WC

19.4.1.5 PORT A Mask Registers (PAMSK,0x20)

PORT A GROUP mask registers.

	PAMSK																													0x20
Bit	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
	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK09	MSK08	MSK07	MSK06	MSK05	MSK04	MSK03	MSK02	MSK01	MSK00
RST	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Name	Description	R/W
31:0	MSK n	Where n = 0 ~ 31 and MSK n = MSK31 ~ MSK0. When INT n = 1: 0: Enable the pin as an interrupt source 1: Disable the pin as an interrupt source When INT n = 0: 0: Corresponding pin will be used as device function 1: Corresponding pin will be used as gpio	RW

19.4.1.6 PORT A Mask Set Registers (PAMSKS,0x24)

PORT A GROUP mask set registers.

	PAMSKS																													0x24
Bit	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
	MSKS31	MSKS30	MSKS29	MSKS28	MSKS27	MSKS26	MSKS25	MSKS24	MSKS23	MSKS22	MSKS21	MSKS20	MSKS19	MSKS18	MSKS17	MSKS16	MSKS15	MSKS14	MSKS13	MSKS12	MSKS11	MSKS10	MSKS09	MSKS08	MSKS07	MSKS06	MSKS05	MSKS04	MSKS03	MSKS02	MSKS01	MSKS00
RST	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

Bits	Name	Description	R/W
31:0	MSKS n	Writing 1 to MSKS n will set MSK n to 1 in register PAMSK. Writing 0 to MSKS n will no use.	WC

19.4.1.7 PORT A Mask Clear Registers (PAMSKC,0x28)

PORT A GROUP mask clear registers.

	PAMSKC																													0x28
Bit	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
	MSKC31	MSKC30	MSKC29	MSKC28	MSKC27	MSKC26	MSKC25	MSKC24	MSKC23	MSKC22	MSKC21	MSKC20	MSKC19	MSKC18	MSKC17	MSKC16	MSKC15	MSKC14	MSKC13	MSKC12	MSKC11	MSKC10	MSKC09	MSKC08	MSKC07	MSKC06	MSKC05	MSKC04	MSKC03	MSKC02	MSKC01	MSKC00
RST	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

Bits	Name	Description	R/W
31:0	MSKC n	Writing 1 to MSKC n will set MSK n to 0 in register PAMSK. Writing 0 to MSKC n will no use.	WC

19.4.1.8 PORT A PAT1/Direction Registers (PAPAT1,0x30)

PORT A GROUP pattern1/direction registers.

	PAPAT1																													0x30
Bit	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
	PAT131	PAT130	PAT129	PAT128	PAT127	PAT126	PAT125	PAT124	PAT123	PAT122	PAT121	PAT120	PAT119	PAT118	PAT117	PAT116	PAT115	PAT114	PAT113	PAT112	PAT111	PAT110	PAT109	PAT108	PAT107	PAT106	PAT105	PAT104	PAT103	PAT102	PAT101	PAT100
RST	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Name	Description	R/W
31:0	PAT1 n	Where n = 0 ~ 31 and PAPAT1 n = PAT131 ~ PAT10. When INT n = 1 (Interrupt function): 0: Level trigger interrupt 1: Edge trigger interrupt When INT n = 0 and MSK = 0 (Device function): 0: Corresponding pin is used as device 0 or device 1 function 1: Corresponding pin is used as device 2 or device 3 function When INT n = 0 and MSK = 1 (GPIO function): 0: Corresponding pin is used as gpio output 1: Corresponding pin is used as gpio input	RW

19.4.1.9 PORT A PAT1/Direction Set Registers (PAPAT1S,0x34)

PORT A GROUP pattern1 or direction set registers.

	PAPAT1S																													0x34
Bit	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
	PAT1S31	PAT1S30	PAT1S29	PAT1S28	PAT1S27	PAT1S26	PAT1S25	PAT1S24	PAT1S23	PAT1S22	PAT1S21	PAT1S20	PAT1S19	PAT1S18	PAT1S17	PAT1S16	PAT1S15	PAT1S14	PAT1S13	PAT1S12	PAT1S11	PAT1S10	PAT1S09	PAT1S08	PAT1S07	PAT1S06	PAT1S05	PAT1S04	PAT1S03	PAT1S02	PAT1S01	PAT1S00
RST	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

Bits	Name	Description	R/W
31:0	PAT1S n	Writing 1 to PAT1S n will set PAT1 n to 1 in register PAPAT1. Writing 0 to PAT1S n will no use.	WC

19.4.1.10 PORT A PAT1/Direction Clear Registers (PAPAT1C,0x38)

PORT A GROUP pattern1 or direction clear registers.

	PAPAT1S,																													0x38
Bit	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
	PAT1C31	PAT1C30	PAT1C29	PAT1C28	PAT1C27	PAT1C26	PAT1C25	PAT1C24	PAT1C23	PAT1C22	PAT1C21	PAT1C20	PAT1C19	PAT1C18	PAT1C17	PAT1C16	PAT1C15	PAT1C14	PAT1C13	PAT1C12	PAT1C11	PAT1C10	PAT1C09	PAT1C08	PAT1C07	PAT1C06	PAT1C05	PAT1C04	PAT1C03	PAT1C02	PAT1C01	PAT1C00
RST	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

Bits	Name	Description	R/W
31:0	PAT1C n	Writing 1 to PAT1C n will set PAT1 n to 0 in register PAPAT1. Writing 0 to PAT1C n will no use.	WC

19.4.1.11 PORT A PAT0/Data Registers (PAPAT0,0x40)

PORT A GROUP pattern 0 or data registers.

	PAPAT0,																													0x40
Bit	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
	PAT031	PAT030	PAT029	PAT028	PAT027	PAT026	PAT025	PAT024	PAT023	PAT022	PAT021	PAT020	PAT019	PAT018	PAT017	PAT016	PAT015	PAT014	PAT013	PAT012	PAT011	PAT010	PAT009	PAT008	PAT007	PAT006	PAT005	PAT004	PAT003	PAT002	PAT001	PAT000
RST	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

Bits	Name	Description	R/W
31:0	PAT0 n	Where n = 0 ~ 31 and PAT0 n = PAT00 ~ PAT031. When INTn = 1 and PAT1 = 0: 0: Port is low level triggered interrupt input 1: Port is low high triggered interrupt input When INTn = 1 and PAT1 = 1: 0: Port is falling edge triggered interrupt input 1: Port is rising edge triggered interrupt input When INTn = 0 and MSK = 0 and PAT1 = 0: 0: Port is pin of device 0 1: Port is pin of device 1 When INTn = 0 and MSK = 0 and PAT1 = 1: 0: Port is pin of device 2 1: Port is pin of device 3 When INTn = 0 and MSK = 1 and PAT1 = 0: 0: Port is GPIO output 0 1: Port is GPIO output 1	RW

19.4.1.12 PORT A PAT0/Data Set Registers (PAPAT0S,0x44)

PORT A GROUP pattern0 or data set registers.

	PAPAT0S																													0x44
Bit	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
	PAT0S31	PAT0S30	PAT0S29	PAT0S28	PAT0S27	PAT0S26	PAT0S25	PAT0S24	PAT0S23	PAT0S22	PAT0S21	PAT0S20	PAT0S19	PAT0S18	PAT0S17	PAT0S16	PAT0S15	PAT0S14	PAT0S13	PAT0S12	PAT0S11	PAT0S10	PAT0S09	PAT0S08	PAT0S07	PAT0S06	PAT0S05	PAT0S04	PAT0S03	PAT0S02	PAT0S01	PAT0S00
RST	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

Bits	Name	Description	R/W
31:0	PAT0S n	Writing 1 to PAPAT0S n will set PAT0 n to 1 in register PAPAT0. Writing 0 to PAPAT0S n will no use.	WC

19.4.1.13 PORT A PAT0/Data Clear Registers (PAPAT0C,0x48)

PORT A GROUP pattern 0 or data clear registers.

	PAPAT0C																													0x48
Bit	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
	PAT0C31	PAT0C30	PAT0C29	PAT0C28	PAT0C27	PAT0C26	PAT0C25	PAT0C24	PAT0C23	PAT0C22	PAT0C21	PAT0C20	PAT0C19	PAT0C18	PAT0C17	PAT0C16	PAT0C15	PAT0C14	PAT0C13	PAT0C12	PAT0C11	PAT0C10	PAT0C09	PAT0C08	PAT0C07	PAT0C06	PAT0C05	PAT0C04	PAT0C03	PAT0C02	PAT0C01	PAT0C00
RST	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

Bits	Name	Description	R/W
31:0	PAT0C n	Writing 1 to PAT0C n will set PAT0 n to 0 in register PAPAT0. Writing 0 to PAT0C n will no use.	WC

19.4.1.14 PORT A FLAG Registers (PAFLG,0x50)

PORT A GROUP flag registers. They are read-only registers.

	PAFLG																													0x50
Bit	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
	FLAG31	FLAG30	FLAG29	FLAG28	FLAG27	FLAG26	FLAG25	FLAG24	FLAG23	FLAG22	FLAG21	FLAG20	FLAG19	FLAG18	FLAG17	FLAG16	FLAG15	FLAG14	FLAG13	FLAG12	FLAG11	FLAG10	FLAG09	FLAG08	FLAG07	FLAG06	FLAG05	FLAG04	FLAG03	FLAG02	FLAG01	FLAG00
RST	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

Bits	Name	Description	R/W
31:0	FLAG n	Where n = 0 ~ 31 and FLAG n = FLAG00 ~ FLAG31. FLAG n is interrupt flag bit for checking the interrupt whether to happen. When GPIO is used as interrupt function and the interrupt happened, the FLAG n in PAFLG will be set to 1.	R

19.4.1.15 PORT A FLAG Clear Registers (PAFLGC,0x58)

PORT A GROUP flag clear registers. They are read-only registers.

	PAFLGC																													0x58
Bit	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
	FLAGC31	FLAGC30	FLAGC29	FLAGC28	FLAGC27	FLAGC26	FLAGC25	FLAGC24	FLAGC23	FLAGC22	FLAGC21	FLAGC20	FLAGC19	FLAGC18	FLAGC17	FLAGC16	FLAGC15	FLAGC14	FLAGC13	FLAGC12	FLAGC11	FLAGC10	FLAGC09	FLAGC08	FLAGC07	FLAGC06	FLAGC05	FLAGC04	FLAGC03	FLAGC02	FLAGC01	FLAGC00
RST	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

Bits	Name	Description	R/W
31:0	FLAGC n	When GPIO is used as interrupt function and when write 1 to the bit, the bit FLAG n in PAFLG will be cleared.	RC

19.4.1.16 PORT A PULL Disable Registers (PAPEN,0x70)

PORT A GROUP pull disable registers.

	PAPEN																													0x70
Bit	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
	PEN31	PEN30	PEN29	PEN28	PEN27	PEN26	PEN25	PEN24	PEN23	PEN22	PEN21	PEN20	PEN19	PEN18	PEN17	PEN16	PEN15	PEN14	PEN13	PEN12	PEN11	PEN10	PEN09	PEN08	PEN07	PEN06	PEN05	PEN04	PEN03	PEN02	PEN01	PEN00
RST	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

Bits	Name	Description	R/W
31:0	PEN n	Where n = 0 ~ 31 and PEN n = PEN0 ~ PEN31. PEN n is used for setting the port to be PULL function enable. 0: port PULL enable 1: port PULL disable	RW

19.4.1.17 PORT A PULL Set Registers (PAPENS,0x74)

PORT A GROUP pull set registers. They are write-only registers.

	PAPENS																													0x74
Bit	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
	PENS31	PENS30	PENS29	PENS28	PENS27	PENS26	PENS25	PENS24	PENS23	PENS22	PENS21	PENS20	PENS19	PENS18	PENS17	PENS16	PENS15	PENS14	PENS13	PENS12	PENS11	PENS10	PENS09	PENS08	PENS07	PENS06	PENS05	PENS04	PENS03	PENS02	PENS01	PENS00
RST	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

Bits	Name	Description	R/W
31:0	PENS n	Writing 1 to PENS n will set PEN n to 1 in register PAPEN. Writing 0 to PENS n will no use.	WC

19.4.1.18 PORT A PULL Clear Registers (PAPENC,0x78)

PORT A pull clear registers. They are write-only registers.

	PAPENC																													0x78
Bit	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
	PENC31	PENC30	PENC29	PENC28	PENC27	PENC26	PENC25	PENC24	PENC23	PENC22	PENC21	PENC20	PENC19	PENC18	PENC17	PENC16	PENC15	PENC14	PENC13	PENC12	PENC11	PENC10	PENC09	PENC08	PENC07	PENC06	PENC05	PENC04	PENC03	PENC02	PENC01	PENC00
RST	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

Bits	Name	Description	R/W
31:0	PENC n	Writing 1 to PENC n will set PEN n to 0 in register PAPEN. Writing 0 to PENC n will no use.	WC

19.4.2 PORT B Register Group

19.4.2.1 PORT B PIN Level Registers (PBPINL,0x00)

PORT B GROUP level registers. They are read-only registers.

	PBPINL																														0x00
Bit	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
	PINL31	PINL30	PINL29	PINL28	PINL27	PINL26	PINL25	PINL24	PINL23	PINL22	PINL21	PINL20	PINL19	PINL18	PINL17	PINL16	PINL15	PINL14	PINL13	PINL12	PINL11	PINL10	PINL09	PINL08	PINL07	PINL06	PINL05	PINL04	PINL03	PINL02	PINL01	PINL00
RST	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

Bits	Name	Description	R/W
31:0	PINL n	Where n = 0 ~ 31 and PINL n = PINL0 ~ PINL31. The PORT PIN level can be read by reading PINL n bit in register PBPINL.	R

19.4.2.2 PORT B Interrupt Registers (PBINT,0x10)

PORT B GROUP interrupt registers.

	PBINT																														0x10
Bit	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
	INT31	INT30	INT29	INT28	INT27	INT26	INT25	INT24	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16	INT15	INT14	INT13	INT12	INT11	INT10	INT09	INT08	INT07	INT06	INT05	INT04	INT03	INT02	INT01	INT00
RST	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

Bits	Name	Description	R/W
31:0	INT n	Where n = 0 ~ 31 and INT n = INT31 ~ INT00. Interrupt enable. 0: Corresponding pin is used as device functions or normal gpio 1: Corresponding pin is used as interrupt	RW

19.4.2.3 PORT B Interrupt Set Registers (PBINTS,0x14)

PORT B GROUP interrupt set registers.

	PBINTS																													0x14
Bit	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
	INTS31	INTS30	INTS29	INTS28	INTS27	INTS26	INTS25	INTS24	INTS23	INTS22	INTS21	INTS20	INTS19	INTS18	INTS17	INTS16	INTS15	INTS14	INTS13	INTS12	INTS11	INTS10	INTS09	INTS08	INTS07	INTS06	INTS05	INTS04	INTS03	INTS02	INTS01	INTS00
RST	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

Bits	Name	Description	R/W
31:0	INTS n	Writing 1 to INTS n will set INT n to 1 in register PBINT. Writing 0 to INTS n will no use.	WC

19.4.2.4 PORT B Interrupt Clear Registers (PBINTC,0x18)

PORT B GROUP interrupt clear registers.

	PBINTC																													0x18
Bit	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
	INTC31	INTC30	INTC29	INTC28	INTC27	INTC26	INTC25	INTC24	INTC23	INTC22	INTC21	INTC20	INTC19	INTC18	INTC17	INTC16	INTC15	INTC14	INTC13	INTC12	INTC11	INTC10	INTC09	INTC08	INTC07	INTC06	INTC05	INTC04	INTC03	INTC02	INTC01	INTC00
RST	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

Bits	Name	Description	R/W
31:0	INTC n	Writing 1 to INTC n will set INT n to 0 in register PBINT. Writing 0 to INTC n will no use.	WC

19.4.2.5 PORT B Mask Registers (PBMSK,0x20)

PORT B GROUP mask registers.

	PBMSK																													0x20
Bit	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
	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK09	MSK08	MSK07	MSK06	MSK05	MSK04	MSK03	MSK02	MSK01	MSK00
RST	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Name	Description	R/W
31:0	MSK n	Where n = 0 ~ 31 and MSK n = MSK31 ~ MSK0. When INT n = 1: 0: Enable the pin as an interrupt source 1: Disable the pin as an interrupt source When INT n = 0: 0: Corresponding pin will be used as device function 1: Corresponding pin will be used as gpio	RW

19.4.2.6 PORT B Mask Set Registers (PBMSKS,0x24)

PORT B GROUP mask set registers.

	PBMSKS																												0x24
Bit	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
	MSKS31	MSKS30	MSKS29	MSKS28	MSKS27	MSKS26	MSKS25	MSKS24	MSKS23	MSKS22	MSKS21	MSKS20	MSKS19	MSKS18	MSKS17	MSKS16	MSKS15	MSKS14	MSKS13	MSKS12	MSKS11	MSKS10	MSKS09	MSKS08	MSKS07	MSKS06	MSKS05	MSKS04	MSKS03	MSKS02	MSKS01	MSKS00
RST	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

Bits	Name	Description	R/W
31:0	MSKS n	Writing 1 to MSKS n will set MSK n to 1 in register PBMSK. Writing 0 to MSKS n will no use.	WC

19.4.2.7 PORT B Mask Clear Registers (PBMSKC,0x28)

PORT B GROUP mask clear registers.

	PBMSKC																												0x28
Bit	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
	MSKC31	MSKC30	MSKC29	MSKC28	MSKC27	MSKC26	MSKC25	MSKC24	MSKC23	MSKC22	MSKC21	MSKC20	MSKC19	MSKC18	MSKC17	MSKC16	MSKC15	MSKC14	MSKC13	MSKC12	MSKC11	MSKC10	MSKC09	MSKC08	MSKC07	MSKC06	MSKC05	MSKC04	MSKC03	MSKC02	MSKC01	MSKC00
RST	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

Bits	Name	Description	R/W
31:0	MSKC n	Writing 1 to MSKC n will set MSK n to 0 in register PBMSK. Writing 0 to MSKC n will no use.	WC

19.4.2.8 PORT B PAT1/Direction Registers (PBPAT1,0x30)

PORT B GROUP pattern1/direction registers.

	PBPAT1																													0x30
Bit	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
	PAT131	PAT130	PAT129	PAT128	PAT127	PAT126	PAT125	PAT124	PAT123	PAT122	PAT121	PAT120	PAT119	PAT118	PAT117	PAT116	PAT115	PAT114	PAT113	PAT112	PAT111	PAT110	PAT109	PAT108	PAT107	PAT106	PAT105	PAT104	PAT103	PAT102	PAT101	PAT100
RST	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Name	Description	R/W
31:0	PAT1 n	Where n = 0 ~ 31 and PAT1 n = PAT131 ~ PAT10. When INT n = 1 (Interrupt function): 0: Level trigger interrupt 1: Edge trigger interrupt When INT n = 0 and PMSK = 0 (Device function): 0: Corresponding pin is used as device 0 or device 1 function 1: Corresponding pin is used as device 2 or device 3 function When INT n = 0 and PMSK = 1 (GPIO function): 0: Corresponding pin is used as gpio output 1: Corresponding pin is used as gpio input	RW

19.4.2.9 PORT B PAT1/Direction Set Registers (PBPAT1S,0x34)

PORT B GROUP pattern1 or direction set registers.

	PBPAT1S																													0x34
Bit	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
	PAT1S31	PAT1S30	PAT1S29	PAT1S28	PAT1S27	PAT1S26	PAT1S25	PAT1S24	PAT1S23	PAT1S22	PAT1S21	PAT1S20	PAT1S19	PAT1S18	PAT1S17	PAT1S16	PAT1S15	PAT1S14	PAT1S13	PAT1S12	PAT1S11	PAT1S10	PAT1S09	PAT1S08	PAT1S07	PAT1S06	PAT1S05	PAT1S04	PAT1S03	PAT1S02	PAT1S01	PAT1S00
RST	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

Bits	Name	Description	R/W
31:0	PAT1S n	Writing 1 to PAT1S n will set PAT1 n to 1 in register PBPAT1. Writing 0 to PAT1S n will no use.	WC

19.4.2.10 PORT B PAT1/Direction Clear Registers (PBPAT1C,0x38)

PORT B GROUP pattern1 or direction clear registers.

	PBPAT1C																													0x38
Bit	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
	PAT1C31	PAT1C30	PAT1C29	PAT1C28	PAT1C27	PAT1C26	PAT1C25	PAT1C24	PAT1C23	PAT1C22	PAT1C21	PAT1C20	PAT1C19	PAT1C18	PAT1C17	PAT1C16	PAT1C15	PAT1C14	PAT1C13	PAT1C12	PAT1C11	PAT1C10	PAT1C09	PAT1C08	PAT1C07	PAT1C06	PAT1C05	PAT1C04	PAT1C03	PAT1C02	PAT1C01	PAT1C00
RST	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

Bits	Name	Description	R/W
31:0	PAT1C n	Writing 1 to PAT1C n will set PAT1 n to 0 in register PBPAT1. Writing 0 to PAT1C n will no use.	WC

19.4.2.11 PORT B PAT0/Data Registers (PBPAT0,0x40)

PORT B GROUP pattern 0 or data registers.

	PBPAT0,																													0x40
Bit	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
	PAT031	PAT030	PAT029	PAT028	PAT027	PAT026	PAT025	PAT024	PAT023	PAT022	PAT021	PAT020	PAT019	PAT018	PAT017	PAT016	PAT015	PAT014	PAT013	PAT012	PAT011	PAT010	PAT009	PAT008	PAT007	PAT006	PAT005	PAT004	PAT003	PAT002	PAT001	PAT000
RST	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

Bits	Name	Description	R/W
31:0	PAT0 n	Where n = 0 ~ 31 and PAT0 n = PAT00 ~ PAT031. When INTn = 1 and PAT1 = 0: 0: Port is low level triggered interrupt input 1: Port is low high triggered interrupt input When INTn = 1 and PAT1 = 1: 0: Port is falling edge triggered interrupt input 1: Port is rising edge triggered interrupt input When INTn = 0 and MSK = 0 and PAT1 = 0: 0: Port is pin of device 0 1: Port is pin of device 1 When INTn = 0 and MSK = 0 and PAT1 = 1: 0: Port is pin of device 2 1: Port is pin of device 3 When INTn = 0 and MSK = 1 and PAT1 = 0: 0: Port is GPIO output 0 1: Port is GPIO output 1	RW

19.4.2.12 PORT B PAT0/Data Set Registers (PBPAT0S,0x44)

PORT B GROUP pattern0 or data set registers.

	PBPAT0S																													0x44
Bit	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
	PAT0S31	PAT0S30	PAT0S29	PAT0S28	PAT0S27	PAT0S26	PAT0S25	PAT0S24	PAT0S23	PAT0S22	PAT0S21	PAT0S20	PAT0S19	PAT0S18	PAT0S17	PAT0S16	PAT0S15	PAT0S14	PAT0S13	PAT0S12	PAT0S11	PAT0S10	PAT0S09	PAT0S08	PAT0S07	PAT0S06	PAT0S05	PAT0S04	PAT0S03	PAT0S02	PAT0S01	PAT0S00
RST	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

Bits	Name	Description	R/W
31:0	PAT0S n	Writing 1 to PAT0S n will set PAT0 n to 1 in register PBPAT0. Writing 0 to PAT0S n will no use.	WC

19.4.2.13 PORT B PAT0/Data Clear Registers (PBPAT0C,0x48)

PORT B GROUP pattern 0 or data clear registers.

	PBPAT0C																													0x48
Bit	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
	PAT0C31	PAT0C30	PAT0C29	PAT0C28	PAT0C27	PAT0C26	PAT0C25	PAT0C24	PAT0C23	PAT0C22	PAT0C21	PAT0C20	PAT0C19	PAT0C18	PAT0C17	PAT0C16	PAT0C15	PAT0C14	PAT0C13	PAT0C12	PAT0C11	PAT0C10	PAT0C09	PAT0C08	PAT0C07	PAT0C06	PAT0C05	PAT0C04	PAT0C03	PAT0C02	PAT0C01	PAT0C00
RST	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

Bits	Name	Description	R/W
31:0	PAT0C n	Writing 1 to PAT0C n will set PAT0 n to 0 in register PBPAT0. Writing 0 to PAT0C n will no use.	WC

19.4.2.14 PORT B FLAG Registers (PBFLG,0x50)

PORT B GROUP flag registers. They are read-only registers.

	PBFLG																													0x50
Bit	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
	FLAG31	FLAG30	FLAG29	FLAG28	FLAG27	FLAG26	FLAG25	FLAG24	FLAG23	FLAG22	FLAG21	FLAG20	FLAG19	FLAG18	FLAG17	FLAG16	FLAG15	FLAG14	FLAG13	FLAG12	FLAG11	FLAG10	FLAG09	FLAG08	FLAG07	FLAG06	FLAG05	FLAG04	FLAG03	FLAG02	FLAG01	FLAG00
RST	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

Bits	Name	Description	R/W
31:0	FLAG n	Where n = 0 ~ 31 and FLAG n = FLAG00 ~ FLAG31. FLAG n is interrupt flag bit for checking the interrupt whether to happen. When GPIO is used as interrupt function and the interrupt happened, the FLAG n in PBFLG will be set to 1.	R

19.4.2.15 PORT B FLAG Clear Registers (PBFLGC,0x58)

PORT B GROUP flag clear registers. They are read-only registers.

	PBFLGC																													0x58
Bit	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
	FLAGC31	FLAGC30	FLAGC29	FLAGC28	FLAGC27	FLAGC26	FLAGC25	FLAGC24	FLAGC23	FLAGC22	FLAGC21	FLAGC20	FLAGC19	FLAGC18	FLAGC17	FLAGC16	FLAGC15	FLAGC14	FLAGC13	FLAGC12	FLAGC11	FLAGC10	FLAGC09	FLAGC08	FLAGC07	FLAGC06	FLAGC05	FLAGC04	FLAGC03	FLAGC02	FLAGC01	FLAGC00
RST	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

Bits	Name	Description	R/W
31:0	FLAGC n	When GPIO is used as interrupt function and when write 1 to the bit, the bit FLAG n in PBFLG will be cleared.	RC

19.4.2.16 PORT B PULL Disable Registers (PBPEN,0x70)

PORT B GROUP pull disable registers.

	PBPEN																													0x70
Bit	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
	PEN31	PEN30	PEN29	PEN28	PEN27	PEN26	PEN25	PEN24	PEN23	PEN22	PEN21	PEN20	PEN19	PEN18	PEN17	PEN16	PEN15	PEN14	PEN13	PEN12	PEN11	PEN10	PEN09	PEN08	PEN07	PEN06	PEN05	PEN04	PEN03	PEN02	PEN01	PEN00
RST	0	1	1	1	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

Bits	Name	Description	R/W
31:0	PEN n	Where n = 0 ~ 31 and PEN n = PEN0 ~ PEN31. PEN n is used for setting the port to be PULL function enable. 0: port PULL enable 1: port PULL disable	RW

19.4.2.17 PORT B PULL Set Registers (PBPENS,0x74)

PORT B GROUP pull set registers. They are write-only registers.

	PBPENS																													0x74
Bit	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
	PENS31	PENS30	PENS29	PENS28	PENS27	PENS26	PENS25	PENS24	PENS23	PENS22	PENS21	PENS20	PENS19	PENS18	PENS17	PENS16	PENS15	PENS14	PENS13	PENS12	PENS11	PENS10	PENS09	PENS08	PENS07	PENS06	PENS05	PENS04	PENS03	PENS02	PENS01	PENS00
RST	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

Bits	Name	Description	R/W
31:0	PENS n	Writing 1 to PENS n will set PEN n to 1 in register PBPEN. Writing 0 to PENS n will no use.	WC

19.4.2.18 PORT B PULL Clear Registers (PBPENC,0x78)

PORT pull clear registers. They are write-only registers.

	PBPENC																													0x78
Bit	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
	PENC31	PENC30	PENC29	PENC28	PENC27	PENC26	PENC25	PENC24	PENC23	PENC22	PENC21	PENC20	PENC19	PENC18	PENC17	PENC16	PENC15	PENC14	PENC13	PENC12	PENC11	PENC10	PENC09	PENC08	PENC07	PENC06	PENC05	PENC04	PENC03	PENC02	PENC01	PENC00
RST	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

Bits	Name	Description	R/W
31:0	PENC n	Writing 1 to PENC n will set PEN n to 0 in register PBPEN. Writing 0 to PENC n will no use.	WC

19.4.3 PORT C Register Group

19.4.3.1 PORT C PIN Level Registers (PCPINL,0x00)

PORT C GROUP level registers. They are read-only registers.

	PCPINL																													0x00
Bit	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
	PINL31	PINL30	PINL29	PINL28	PINL27	PINL26	PINL25	PINL24	PINL23	PINL22	PINL21	PINL20	PINL19	PINL18	PINL17	PINL16	PINL15	PINL14	PINL13	PINL12	PINL11	PINL10	PINL09	PINL08	PINL07	PINL06	PINL05	PINL04	PINL03	PINL02	PINL01	PINL00
RST	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

Bits	Name	Description	R/W
31:0	PINL n	Where n = 0 ~ 31 and PINL n = PINL0 ~ PINL31. The PORT PIN level can be read by reading PINL n bit in register PCPIN.	R

19.4.3.2 PORT C Interrupt Registers (PCINT,0x10)

PORT C GROUP interrupt registers.

	PCINT																													0x10
Bit	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
	INT31	INT30	INT29	INT28	INT27	INT26	INT25	INT24	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16	INT15	INT14	INT13	INT12	INT11	INT10	INT09	INT08	INT07	INT06	INT05	INT04	INT03	INT02	INT01	INT00
RST	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

Bits	Name	Description	R/W
31:0	INT n	Where n = 0 ~ 31 and INT n = INT31 ~ INT00. Interrupt enable. 0: Corresponding pin is used as device functions or normal gpio 1: Corresponding pin is used as interrupt	RW

19.4.3.3 PORT C Interrupt Set Registers (PCINTS,0x14)

PORT C GROUP interrupt set registers.

	PCINTS																													0x14
Bit	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
	INTS31	INTS30	INTS29	INTS28	INTS27	INTS26	INTS25	INTS24	INTS23	INTS22	INTS21	INTS20	INTS19	INTS18	INTS17	INTS16	INTS15	INTS14	INTS13	INTS12	INTS11	INTS10	INTS09	INTS08	INTS07	INTS06	INTS05	INTS04	INTS03	INTS02	INTS01	INTS00
RST	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

Bits	Name	Description	R/W
31:0	INTS n	Writing 1 to INTS n will set INT n to 1 in register PCINT. Writing 0 to INTS n will no use.	WC

19.4.3.4 PORT C Interrupt Clear Registers (PCINTC,0x18)

PORT C GROUP interrupt clear registers.

	PCINTC																													0x18
Bit	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
	INTC31	INTC30	INTC29	INTC28	INTC27	INTC26	INTC25	INTC24	INTC23	INTC22	INTC21	INTC20	INTC19	INTC18	INTC17	INTC16	INTC15	INTC14	INTC13	INTC12	INTC11	INTC10	INTC09	INTC08	INTC07	INTC06	INTC05	INTC04	INTC03	INTC02	INTC01	INTC00
RST	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

Bits	Name	Description	R/W
31:0	INTC n	Writing 1 to INTC n will set INT n to 0 in register PCINT. Writing 0 to INTC n will no use.	WC

19.4.3.5 PORT C Mask Registers (PCMSK,0x20)

PORT C GROUP mask registers.

	PCMSK																													0x20
Bit	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
	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK09	MSK08	MSK07	MSK06	MSK05	MSK04	MSK03	MSK02	MSK01	MSK00
RST	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Name	Description	R/W
31:0	MSK n	Where n = 0 ~ 31 and MSK n = MSK31 ~ MSK0. When INT n = 1: 0: Enable the pin as an interrupt source 1: Disable the pin as an interrupt source When INT n = 0: 0: Corresponding pin will be used as device function 1: Corresponding pin will be used as gpio	RW

19.4.3.6 PORT C Mask Set Registers (PCMSKS,0x24)

PORT C GROUP mask set registers.

	PCMSKS																													0x24
Bit	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
	MSKS31	MSKS30	MSKS29	MSKS28	MSKS27	MSKS26	MSKS25	MSKS24	MSKS23	MSKS22	MSKS21	MSKS20	MSKS19	MSKS18	MSKS17	MSKS16	MSKS15	MSKS14	MSKS13	MSKS12	MSKS11	MSKS10	MSKS09	MSKS08	MSKS07	MSKS06	MSKS05	MSKS04	MSKS03	MSKS02	MSKS01	MSKS00
RST	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

Bits	Name	Description	R/W
31:0	MSKS n	Writing 1 to MSKS n will set MSK n to 1 in register PCMSK. Writing 0 to MSKS n will no use.	WC

19.4.3.7 PORT C Mask Clear Registers (PCMSKC,0x28)

PORT C GROUP mask clear registers.

	PCMSKC																													0x28
Bit	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
	MSKC31	MSKC30	MSKC29	MSKC28	MSKC27	MSKC26	MSKC25	MSKC24	MSKC23	MSKC22	MSKC21	MSKC20	MSKC19	MSKC18	MSKC17	MSKC16	MSKC15	MSKC14	MSKC13	MSKC12	MSKC11	MSKC10	MSKC09	MSKC08	MSKC07	MSKC06	MSKC05	MSKC04	MSKC03	MSKC02	MSKC01	MSKC00
RST	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

Bits	Name	Description	R/W
31:0	MSKC n	Writing 1 to MSKC n will set MSK n to 0 in register PCMSK. Writing 0 to MSKC n will no use.	WC

19.4.3.8 PORT C PAT1/Direction Registers (PCPAT1,0x30)

PORT C GROUP pattern1/direction registers.

	PCPAT1																													0x30
Bit	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
	PAT131	PAT130	PAT129	PAT128	PAT127	PAT126	PAT125	PAT124	PAT123	PAT122	PAT121	PAT120	PAT119	PAT118	PAT117	PAT116	PAT115	PAT114	PAT113	PAT112	PAT111	PAT110	PAT109	PAT108	PAT107	PAT106	PAT105	PAT104	PAT103	PAT102	PAT101	PAT100
RST	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Name	Description	R/W
31:0	PAT1 n	Where n = 0 ~ 31 and PAT1 n = PAT131 ~ PAT10. When INT n = 1 (Interrupt function): 0: Level trigger interrupt 1: Edge trigger interrupt When INT n = 0 and MSK = 0 (Device function): 0: Corresponding pin is used as device 0 or device 1 function 1: Corresponding pin is used as device 2 or device 3 function When INT n = 0 and MSK = 1 (GPIO function): 0: Corresponding pin is used as gpio output 1: Corresponding pin is used as gpio input	RW

19.4.3.9 PORT C PAT1/Direction Set Registers (PCPAT1S,0x34)

PORT C GROUP pattern1 or direction set registers.

	PCPAT1S																													0x34
Bit	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
	PAT1S31	PAT1S30	PAT1S29	PAT1S28	PAT1S27	PAT1S26	PAT1S25	PAT1S24	PAT1S23	PAT1S22	PAT1S21	PAT1S20	PAT1S19	PAT1S18	PAT1S17	PAT1S16	PAT1S15	PAT1S14	PAT1S13	PAT1S12	PAT1S11	PAT1S10	PAT1S09	PAT1S08	PAT1S07	PAT1S06	PAT1S05	PAT1S04	PAT1S03	PAT1S02	PAT1S01	PAT1S00
RST	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

Bits	Name	Description	R/W
31:0	PAT1S n	Writing 1 to PAT1S n will set PAT1 n to 1 in register PAPAT1. Writing 0 to PAT1S n will no use.	WC

19.4.3.10 PORT C PAT1/Direction Clear Registers (PCPAT1C,0x38)

PORT C GROUP pattern1 or direction clear registers.

	PCPAT1S,																													0x38
Bit	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
	PAT1C31	PAT1C30	PAT1C29	PAT1C28	PAT1C27	PAT1C26	PAT1C25	PAT1C24	PAT1C23	PAT1C22	PAT1C21	PAT1C20	PAT1C19	PAT1C18	PAT1C17	PAT1C16	PAT1C15	PAT1C14	PAT1C13	PAT1C12	PAT1C11	PAT1C10	PAT1C09	PAT1C08	PAT1C07	PAT1C06	PAT1C05	PAT1C04	PAT1C03	PAT1C02	PAT1C01	PAT1C00
RST	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

Bits	Name	Description	R/W
31:0	PAT1C n	Writing 1 to PAT1C n will set PAT1 n to 0 in register PAPAT1. Writing 0 to PAT1C n will no use.	WC

19.4.3.11 PORT C PAT0/Data Registers (PCPAT0,0x40)

PORT C GROUP pattern 0 or data registers.

	PCPAT0,																													0x40
Bit	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
	PAT031	PAT030	PAT029	PAT028	PAT027	PAT026	PAT025	PAT024	PAT023	PAT022	PAT021	PAT020	PAT019	PAT018	PAT017	PAT016	PAT015	PAT014	PAT013	PAT012	PAT011	PAT010	PAT009	PAT008	PAT007	PAT006	PAT005	PAT004	PAT003	PAT002	PAT001	PAT000
RST	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

Bits	Name	Description	R/W
31:0	PAT0 n	Where n = 0 ~ 31 and PAT0 n = PAT00 ~ PAT031.	RW

		When INTn = 1 and PAT1 = 0: 0: Port is low level triggered interrupt input 1: Port is low high triggered interrupt input When INTn = 1 and PAT1 = 1: 0: Port is falling edge triggered interrupt input 1: Port is rising edge triggered interrupt input When INTn = 0 and MSK = 0 and PAT1 = 0: 0: Port is pin of device 0 1: Port is pin of device 1 When INTn = 0 and MSK = 0 and PAT1 = 1: 0: Port is pin of device 2 1: Port is pin of device 3 When INTn = 0 and MSK = 1 and PAT1 = 0: 0: Port is GPIO output 0 1: Port is GPIO output 1	

19.4.3.12 PORT C PAT0/Data Set Registers (PCPAT0S,0x44)

PORT C GROUP pattern0 or data set registers.

	PCPAT0S																													0x44
Bit	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
	PAT0S31	PAT0S30	PAT0S29	PAT0S28	PAT0S27	PAT0S26	PAT0S25	PAT0S24	PAT0S23	PAT0S22	PAT0S21	PAT0S20	PAT0S19	PAT0S18	PAT0S17	PAT0S16	PAT0S15	PAT0S14	PAT0S13	PAT0S12	PAT0S11	PAT0S10	PAT0S09	PAT0S08	PAT0S07	PAT0S06	PAT0S05	PAT0S04	PAT0S03	PAT0S02	PAT0S01	PAT0S00
RST	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

Bits	Name	Description	R/W
31:0	PAT0S n	Writing 1 to PAT0S n will set PAT0 n to 1 in register PAPAT0. Writing 0 to PAT0S n will no use.	WC

19.4.3.13 PORT C PAT0/Data Clear Registers (PCPAT0C,0x48)

PORT C GROUP pattern 0 or data clear registers.

	PCPAT0C																													0x48
Bit	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
	PAT0C31	PAT0C30	PAT0C29	PAT0C28	PAT0C27	PAT0C26	PAT0C25	PAT0C24	PAT0C23	PAT0C22	PAT0C21	PAT0C20	PAT0C19	PAT0C18	PAT0C17	PAT0C16	PAT0C15	PAT0C14	PAT0C13	PAT0C12	PAT0C11	PAT0C10	PAT0C09	PAT0C08	PAT0C07	PAT0C06	PAT0C05	PAT0C04	PAT0C03	PAT0C02	PAT0C01	PAT0C00
RST	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

Bits	Name	Description	R/W
31:0	PAT0C n	Writing 1 to PAT0C n will set PAT0 n to 0 in register PAPAT0. Writing 0 to PAT0C n will no use.	WC

19.4.3.14 PORT C FLAG Registers (PCFLG,0x50)

PORT C GROUP flag registers. They are read-only registers.

	PCFLG																													0x50
Bit	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
	FLAG31	FLAG30	FLAG29	FLAG28	FLAG27	FLAG26	FLAG25	FLAG24	FLAG23	FLAG22	FLAG21	FLAG20	FLAG19	FLAG18	FLAG17	FLAG16	FLAG15	FLAG14	FLAG13	FLAG12	FLAG11	FLAG10	FLAG09	FLAG08	FLAG07	FLAG06	FLAG05	FLAG04	FLAG03	FLAG02	FLAG01	FLAG00
RST	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

Bits	Name	Description	R/W
31:0	FLAG n	Where n = 0 ~ 31 and FLAG n = FLAG00 ~ FLAG31. FLAG n is interrupt flag bit for checking the interrupt whether to happen. When GPIO is used as interrupt function and the interrupt happened, the FLAG n in PCFLG will be set to 1.	R

19.4.3.15 PORT C FLAG Clear Registers (PCFLGC,0x58)

PORT C GROUP flag clear registers. They are read-only registers.

	PCFLGC																													0x58
Bit	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
	FLAGC31	FLAGC30	FLAGC29	FLAGC28	FLAGC27	FLAGC26	FLAGC25	FLAGC24	FLAGC23	FLAGC22	FLAGC21	FLAGC20	FLAGC19	FLAGC18	FLAGC17	FLAGC16	FLAGC15	FLAGC14	FLAGC13	FLAGC12	FLAGC11	FLAGC10	FLAGC09	FLAGC08	FLAGC07	FLAGC06	FLAGC05	FLAGC04	FLAGC03	FLAGC02	FLAGC01	FLAGC00
RST	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

Bits	Name	Description	R/W
31:0	FLAGC n	When GPIO is used as interrupt function and when write 1 to the bit, the bit FLAG n in PCFLG will be cleared.	RC

19.4.3.16 PORT C PULL Disable Registers (PCPEN,0x70)

PORT C GROUP pull disable registers.

	PCPEN																													0x70
Bit	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
	PEN31	PEN30	PEN29	PEN28	PEN27	PEN26	PEN25	PEN24	PEN23	PEN22	PEN21	PEN20	PEN19	PEN18	PEN17	PEN16	PEN15	PEN14	PEN13	PEN12	PEN11	PEN10	PEN09	PEN08	PEN07	PEN06	PEN05	PEN04	PEN03	PEN02	PEN01	PEN00
RST	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:0	PEN n	Where n = 0 ~ 31 and PEN n = PEN0 ~ PEN31. PEN n is used for setting the port to be PULL function enable. 0: port PULL enable 1: port PULL disable	RW

19.4.3.17 PORT C PULL Set Registers (PCPENS,0x74)

PORT C GROUP pull set registers. They are write-only registers.

	PCPENS																													0x74
Bit	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
	PENS31	PENS30	PENS29	PENS28	PENS27	PENS26	PENS25	PENS24	PENS23	PENS22	PENS21	PENS20	PENS19	PENS18	PENS17	PENS16	PENS15	PENS14	PENS13	PENS12	PENS11	PENS10	PENS09	PENS08	PENS07	PENS06	PENS05	PENS04	PENS03	PENS02	PENS01	PENS00
RST	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

Bits	Name	Description	R/W
31:0	PENS n	Writing 1 to PENS n will set PEN n to 1 in register PCPEN. Writing 0 to PENS n will no use.	WC

19.4.3.18 PORT C PULL Clear Registers (PCPENC,0x78)

PORT pull clear registers. They are write-only registers.

	PCPENC																													0x78
Bit	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
	PENC31	PENC30	PENC29	PENC28	PENC27	PENC26	PENC25	PENC24	PENC23	PENC22	PENC21	PENC20	PENC19	PENC18	PENC17	PENC16	PENC15	PENC14	PENC13	PENC12	PENC11	PENC10	PENC09	PENC08	PENC07	PENC06	PENC05	PENC04	PENC03	PENC02	PENC01	PENC00
RST	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

Bits	Name	Description	R/W
31:0	PENC n	Writing 1 to PENC n will set PEN n to 0 in register PCPEN. Writing 0 to PENC n will no use.	WC

19.4.3.19 PORT C GLITCH FILTER Configure Register 0 (PCGFCFG0,0x400)

PORT C GROUP glitch filter control bit 0 registers.

	PCGFCFG0																													0x800
Bit	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
	GFCFG031	GFCFG030	GFCFG029	GFCFG028	GFCFG027	GFCFG026	GFCFG025	GFCFG024	GFCFG023	GFCFG022	GFCFG021	GFCFG020	GFCFG019	GFCFG018	GFCFG017	GFCFG016	GFCFG015	GFCFG014	GFCFG013	GFCFG012	GFCFG011	GFCFG010	GFCFG009	GFCFG008	GFCFG007	GFCFG006	GFCFG005	GFCFG004	GFCFG003	GFCFG002	GFCFG001	GFCFG000
RST	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

Bits	Name	Description	R/W
31:0	GFCFG0 n	Where n = 0 ~ 31 and GFCFG0 n = GFCFG00 ~ GFCFG031. GFCFG0 n is used for suppress the glitch filter on port signal bit 0 value . 0: glitch filter control signal bit 0 value is 0 1: glitch filter control signal bit 0 value is 1	RW

NOTE: Not every PAD support glitch filter control, please check out Function Overview chapter for details.

19.4.3.20 PORT C GLITCH FILTER Configure Set Register 0 (PCGFCFG0S,0x404)

PORT C GROUP glitch filter control bit 0 set registers.

	PCGFCFG0S																													0x804
Bit	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
	GFCFG0S31	GFCFG0S30	GFCFG0S29	GFCFG0S28	GFCFG0S27	GFCFG0S26	GFCFG0S25	GFCFG0S24	GFCFG0S23	GFCFG0S22	GFCFG0S21	GFCFG0S20	GFCFG0S19	GFCFG0S18	GFCFG0S17	GFCFG0S16	GFCFG0S15	GFCFG0S14	GFCFG0S13	GFCFG0S12	GFCFG0S11	GFCFG0S10	GFCFG0S09	GFCFG0S08	GFCFG0S07	GFCFG0S06	GFCFG0S05	GFCFG0S04	GFCFG0S03	GFCFG0S02	GFCFG0S01	GFCFG0S00
RST	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

Bits	Name	Description	R/W
31:0	GFCFG0S n	Writing 1 to GFCFG0S n will set glitch filter control signal GFCFG0 n to 0 in register PCGFCFG0. Writing 0 to GFCFG0S n will no use.	WC

19.4.3.21 PORT C GLITCH FILTER Configure Clear Register 0 (PCGFCFG0C,0x408)

PORT C GROUP glitch filter control bit 0 clear registers.

	PCGFCFG0C																													0x808
Bit	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
	GFCFG0C31	GFCFG0C30	GFCFG0C29	GFCFG0C28	GFCFG0C27	GFCFG0C26	GFCFG0C25	GFCFG0C24	GFCFG0C23	GFCFG0C22	GFCFG0C21	GFCFG0C20	GFCFG0C19	GFCFG0C18	GFCFG0C17	GFCFG0C16	GFCFG0C15	GFCFG0C14	GFCFG0C13	GFCFG0C12	GFCFG0C11	GFCFG0C10	GFCFG0C09	GFCFG0C08	GFCFG0C07	GFCFG0C06	GFCFG0C05	GFCFG0C04	GFCFG0C03	GFCFG0C02	GFCFG0C01	GFCFG0C00
RST	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

Bits	Name	Description	R/W
31:0	GFCFG0C n	Writing 1 to GFCFG0C n will clear glitch filter control signal GFCFG0 n to 0 in register PCGFCFG0. Writing 0 to GFCFG0C n will no use.	WC

19.4.3.22 PORT C GLITCH FILTER Configure Register 1 (PCGFCFG1,0x410)

PORT C GROUP glitch filter control bit 1 registers.

	PCGFCFG1																													0x810
Bit	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
	GFCFG131	GFCFG130	GFCFG129	GFCFG128	GFCFG127	GFCFG126	GFCFG125	GFCFG124	GFCFG123	GFCFG122	GFCFG121	GFCFG120	GFCFG119	GFCFG118	GFCFG117	GFCFG116	GFCFG115	GFCFG114	GFCFG113	GFCFG112	GFCFG111	GFCFG110	GFCFG109	GFCFG108	GFCFG107	GFCFG106	GFCFG105	GFCFG104	GFCFG103	GFCFG102	GFCFG101	GFCFG100
RST	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

Bits	Name	Description	R/W
31:0	GFCFG1 n	Where n = 0 ~ 31 and GFCFG1 n = GFCFG10 ~ GFCFG131. GFCFG1 n is used for suppress the glitch filter on port signal bit 1 value . 0: glitch filter control signal bit 1 value is 0 1: glitch filter control signal bit 1 value is 1	RW

NOTE:

1. Not every PAD support glitch filter control, please check out Function Overview chapter for details.

19.4.3.23 PORT C GLITCH FILTER Configure Register 1 (PCGFCFG1S,0x414)

PORT C GROUP glitch filter control bit 1 set registers.

	PCGFCFG1S																													0x814
Bit	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
	GFCFG1S31	GFCFG1S30	GFCFG1S29	GFCFG1S28	GFCFG1S27	GFCFG1S26	GFCFG1S25	GFCFG1S24	GFCFG1S23	GFCFG1S22	GFCFG1S21	GFCFG1S20	GFCFG1S19	GFCFG1S18	GFCFG1S17	GFCFG1S16	GFCFG1S15	GFCFG1S14	GFCFG1S13	GFCFG1S12	GFCFG1S11	GFCFG1S10	GFCFG1S09	GFCFG1S08	GFCFG1S07	GFCFG1S06	GFCFG1S05	GFCFG1S04	GFCFG1S03	GFCFG1S02	GFCFG1S01	GFCFG1S00
RST	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

Bits	Name	Description	R/W
31:0	GFCFG1S n	Writing 1 to GFCFG1S n will set glitch filter control signal GFCFG1 n to 0 in register PCGFCFG1. Writing 0 to GFCFG1S n will no use.	WC

19.4.3.24 PORT C GLITCH FILTER Configure Register 1 (PCGFCFG1C,0x418)

PORT C GROUP glitch filter control bit 1 clear registers.

	PCGFCFG1C																													0x818
Bit	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
	GFCFG1C31	GFCFG1C30	GFCFG1C29	GFCFG1C28	GFCFG1C27	GFCFG1C26	GFCFG1C25	GFCFG1C24	GFCFG1C23	GFCFG1C22	GFCFG1C21	GFCFG1C20	GFCFG1C19	GFCFG1C18	GFCFG1C17	GFCFG1C16	GFCFG1C15	GFCFG1C14	GFCFG1C13	GFCFG1C12	GFCFG1C11	GFCFG1C10	GFCFG1C09	GFCFG1C08	GFCFG1C07	GFCFG1C06	GFCFG1C05	GFCFG1C04	GFCFG1C03	GFCFG1C02	GFCFG1C01	GFCFG1C00
RST	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

Bits	Name	Description	R/W
31:0	GFCFG1C n	Writing 1 to GFCFG1C n will clear glitch filter control signal GFCFG1 n to 0 in register PCGFCFG1. Writing 0 to GFCFG1C n will no use.	WC

19.4.3.25 PORT C GLITCH FILTER Configure Register 2 (PCGFCFG2,0x420)

PORT C GROUP glitch filter control bit 2 registers.

	PCGFCFG2																													0x820
Bit	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
	GFCFG231	GFCFG230	GFCFG229	GFCFG228	GFCFG227	GFCFG226	GFCFG225	GFCFG224	GFCFG223	GFCFG222	GFCFG221	GFCFG220	GFCFG219	GFCFG218	GFCFG217	GFCFG216	GFCFG215	GFCFG214	GFCFG213	GFCFG212	GFCFG211	GFCFG210	GFCFG209	GFCFG208	GFCFG207	GFCFG206	GFCFG205	GFCFG204	GFCFG203	GFCFG202	GFCFG201	GFCFG200
RST	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

Bits	Name	Description	R/W
31:0	GFCFG2 n	Where n = 0 ~ 31 and GFCFG2 n = GFCFG20 ~ GFCFG231. GFCFG2 n is used for suppress the glitch filter on port signal bit 2 value . 0: glitch filter control signal bit 2 value is 0 1: glitch filter control signal bit 2 value is 1	RW

NOTE:

1. Not every PAD support glitch filter Control, please check out Function Overview chapter for details.

19.4.3.26 PORT C GLITCH FILTER Configure Set Register 2 (PCGFCFG2S,0x424)

PORT C GROUP glitch filter control bit 2 set registers.

	PCGFCFG2S																													0x824
Bit	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
	GFCFG2S31	GFCFG2S30	GFCFG2S29	GFCFG2S28	GFCFG2S27	GFCFG2S26	GFCFG2S25	GFCFG2S24	GFCFG2S23	GFCFG2S22	GFCFG2S21	GFCFG2S20	GFCFG2S19	GFCFG2S18	GFCFG2S17	GFCFG2S16	GFCFG2S15	GFCFG2S14	GFCFG2S13	GFCFG2S12	GFCFG2S11	GFCFG2S10	GFCFG2S09	GFCFG2S08	GFCFG2S07	GFCFG2S06	GFCFG2S05	GFCFG2S04	GFCFG2S03	GFCFG2S02	GFCFG2S01	GFCFG2S00
RST	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

Bits	Name	Description	R/W
31:0	GFCFG2 S n	Writing 1 to GFCFG2S n will set glitch filter control signal GFCFG2 n to 0 in register PCGFCFG2. Writing 0 to GFCFG2S n will no use.	WC

19.4.3.27 PORT C GLITCH FILTER Configure Clear Register 2 (PCGFCFG2C,0x428)

PORT C GROUP glitch filter control bit 2 clear registers.

	PCGFCFG2C																													0x828
Bit	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
	GFCFG2C31	GFCFG2C30	GFCFG2C29	GFCFG2C28	GFCFG2C27	GFCFG2C26	GFCFG2C25	GFCFG2C24	GFCFG2C23	GFCFG2C22	GFCFG2C21	GFCFG2C20	GFCFG2C19	GFCFG2C18	GFCFG2C17	GFCFG2C16	GFCFG2C15	GFCFG2C14	GFCFG2C13	GFCFG2C12	GFCFG2C11	GFCFG2C10	GFCFG2C09	GFCFG2C08	GFCFG2C07	GFCFG2C06	GFCFG2C05	GFCFG2C04	GFCFG2C03	GFCFG2C02	GFCFG2C01	GFCFG2C00
RST	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

Bits	Name	Description	R/W
31:0	GFCFG2 C n	Writing 1 to GFCFG2C n will clear glitch filter control signal GFCFG2 n to 0 in register PAGFCFG2. Writing 0 to GFCFG2C n will no use.	WC

19.4.3.28 PORT C GLITCH FILTER Configure Register 3 (PCGFCFG3,0x430)

PORT C GROUP glitch filter control bit 3 registers.

	PCGFCFG3																													0x830
Bit	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
	GFCFG331	GFCFG330	GFCFG329	GFCFG328	GFCFG327	GFCFG326	GFCFG325	GFCFG324	GFCFG323	GFCFG322	GFCFG321	GFCFG320	GFCFG319	GFCFG318	GFCFG317	GFCFG316	GFCFG315	GFCFG314	GFCFG313	GFCFG312	GFCFG311	GFCFG310	GFCFG309	GFCFG308	GFCFG307	GFCFG306	GFCFG305	GFCFG304	GFCFG303	GFCFG302	GFCFG301	GFCFG300
RST	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

Bits	Name	Description	R/W
31:0	GFCFG3 n	Where n = 0 ~ 31 and GFCFG3 n = GFCFG30 ~ GFCFG331. GFCFG3 n is used for suppress the glitch filter on port signal bit 3 value . 0: glitch filter control signal bit 3 value is 0 1: glitch filter control signal bit 3 value is 1	RW

NOTE:

1. Not every PAD support glitch filter control, please check out Function Overview chapter for details.

19.4.3.29 PORT C GLITCH FILTER Configure Set Register 3 (PCGFCFG3S,0x434)

PORT C GROUP glitch filter control bit 3 set registers.

	PCGFCFG3S																													0x834
Bit	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
	GFCFG3S31	GFCFG3S30	GFCFG3S29	GFCFG3S28	GFCFG3S27	GFCFG3S26	GFCFG3S25	GFCFG3S24	GFCFG3S23	GFCFG3S22	GFCFG3S21	GFCFG3S20	GFCFG3S19	GFCFG3S18	GFCFG3S17	GFCFG3S16	GFCFG3S15	GFCFG3S14	GFCFG3S13	GFCFG3S12	GFCFG3S11	GFCFG3S10	GFCFG3S09	GFCFG3S08	GFCFG3S07	GFCFG3S06	GFCFG3S05	GFCFG3S04	GFCFG3S03	GFCFG3S02	GFCFG3S01	GFCFG3S00
RST	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

Bits	Name	Description	R/W
31:0	GFCFG3S n	Writing 1 to GFCFG3S n will set glitch filter control signal GFCFG3 n to 0 in register PAGFCFG3. Writing 0 to GFCFG3S n will no use.	WC

19.4.3.30 PORT C GLITCH FILTER Configure Clear Register 3 (PCGFCFG3C,0x438)

PORT C GROUP glitch filter control bit 3 clear registers.

	PCGFCFG3C																													0x838
Bit	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
	GFCFG3S31	GFCFG3S30	GFCFG3S29	GFCFG3S28	GFCFG3S27	GFCFG3S26	GFCFG3S25	GFCFG3S24	GFCFG3S23	GFCFG3S22	GFCFG3S21	GFCFG3S20	GFCFG3S19	GFCFG3S18	GFCFG3S17	GFCFG3S16	GFCFG3S15	GFCFG3S14	GFCFG3S13	GFCFG3S12	GFCFG3S11	GFCFG3S10	GFCFG3S09	GFCFG3S08	GFCFG3S07	GFCFG3S06	GFCFG3S05	GFCFG3S04	GFCFG3S03	GFCFG3S02	GFCFG3S01	GFCFG3S00
RST	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

Bits	Name	Description	R/W
31:0	GFCFG3C n	Writing 1 to GFCFG3C n will clear glitch filter control signal GFCFG3 n to 0 in register PAGFCFG3. Writing 0 to GFCFG3C n will no use.	WC

These four registers used for suppress the glitch on port signal. The range of glitch width is:
1 x pclk ~ 15 x pclk

Bits	Name	Description	R/W
4	GFCFG n	Glitch Filter Configure Register for each PAD, digital filter which mask upto 15 pclk cycle's unstable input signal	WR

GFCFG3	GFCFG2	GFCFG1	GFCFG0	Drive Strength Level
0	0	0	0	(disable)
0	0	0	1	1 pclk
0	0	1	0	2 pclk
0	0	1	1	3
..	..	..	..	..
1	1	1	0	14
1	1	1	1	15 pclk

Note:
1. not each pad supports Glitch Filter Function

Note:
1. pclk is the clock period configure to APB bus.
2. This function only used in interrupt input mode.
3. Not every bit for each port is implement this function. please refer *1.3*.

19.4.4 PORT D Register Group

19.4.4.1 PORT D PIN Level Registers (PDPINL,0x00)

PORT D GROUP level registers. They are read-only registers.

	PDPINL																													0x00
Bit	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
	PINL31	PINL30	PINL29	PINL28	PINL27	PINL26	PINL25	PINL24	PINL23	PINL22	PINL21	PINL20	PINL19	PINL18	PINL17	PINL16	PINL15	PINL14	PINL13	PINL12	PINL11	PINL10	PINL09	PINL08	PINL07	PINL06	PINL05	PINL04	PINL03	PINL02	PINL01	PINL00
RST	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

Bits	Name	Description	R/W
31:0	PINL n	Where n = 0 ~ 31 and PINL n = PINL0 ~ PINL31. The PORT PIN level can be read by reading PINL n bit in register PDPIN.	R

NOTE:

19.4.4.2 PORT D Interrupt Registers (PDINT,0x10)

PORT D GROUP interrupt registers.

	PDINT																													0x10
Bit	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
	INT31	INT30	INT29	INT28	INT27	INT26	INT25	INT24	INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16	INT15	INT14	INT13	INT12	INT11	INT10	INT09	INT08	INT07	INT06	INT05	INT04	INT03	INT02	INT01	INT00
RST	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

Bits	Name	Description	R/W
31:0	INT n	Where n = 0 ~ 31 and INT n = INT31 ~ INT00. Interrupt enable. 0: Corresponding pin is used as device functions or normal gpio 1: Corresponding pin is used as interrupt	RW

19.4.4.3 PORT D Interrupt Set Registers (PDINTS,0x14)

PORT D GROUP interrupt set registers.

	PDINTS																													0x14
Bit	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
	INTS31	INTS30	INTS29	INTS28	INTS27	INTS26	INTS25	INTS24	INTS23	INTS22	INTS21	INTS20	INTS19	INTS18	INTS17	INTS16	INTS15	INTS14	INTS13	INTS12	INTS11	INTS10	INTS09	INTS08	INTS07	INTS06	INTS05	INTS04	INTS03	INTS02	INTS01	INTS00
RST	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

Bits	Name	Description	R/W
31:0	INTS n	Writing 1 to INTS n will set INT n to 1 in register PDINT. Writing 0 to INTS n will no use.	WC

19.4.4.4 PORT D Interrupt Clear Registers (PDINTC,0x18)

PORT D GROUP interrupt clear registers.

	PDINTC																													0x18
Bit	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
	INTC31	INTC30	INTC29	INTC28	INTC27	INTC26	INTC25	INTC24	INTC23	INTC22	INTC21	INTC20	INTC19	INTC18	INTC17	INTC16	INTC15	INTC14	INTC13	INTC12	INTC11	INTC10	INTC09	INTC08	INTC07	INTC06	INTC05	INTC04	INTC03	INTC02	INTC01	INTC00
RST	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

Bits	Name	Description	R/W
31:0	INTC n	Writing 1 to INTC n will set INT n to 0 in register PDINT. Writing 0 to INTC n will no use.	WC

19.4.4.5 PORT D Mask Registers (PDMSK,0x20)

PORT D GROUP mask registers.

	PDMSK																													0x20
Bit	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
	MSK31	MSK30	MSK29	MSK28	MSK27	MSK26	MSK25	MSK24	MSK23	MSK22	MSK21	MSK20	MSK19	MSK18	MSK17	MSK16	MSK15	MSK14	MSK13	MSK12	MSK11	MSK10	MSK09	MSK08	MSK07	MSK06	MSK05	MSK04	MSK03	MSK02	MSK01	MSK00
RST	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	1	1	1	1	1

Bits	Name	Description	R/W
31:0	MSK n	Where n = 0 ~ 31 and MSK n = MSK31 ~ MSK0. When INT n = 1: 0: Enable the pin as an interrupt source 1: Disable the pin as an interrupt source When INT n = 0: 0: Corresponding pin will be used as device function 1: Corresponding pin will be used as gpio	RW

19.4.4.6 PORT D Mask Set Registers (PDMSKS,0x24)

PORT D GROUP mask set registers.

	PDMSKS																												0x24
Bit	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
	MSKS31	MSKS30	MSKS29	MSKS28	MSKS27	MSKS26	MSKS25	MSKS24	MSKS23	MSKS22	MSKS21	MSKS20	MSKS19	MSKS18	MSKS17	MSKS16	MSKS15	MSKS14	MSKS13	MSKS12	MSKS11	MSKS10	MSKS09	MSKS08	MSKS07	MSKS06	MSKS05	MSKS04	MSKS03	MSKS02	MSKS01	MSKS00
RST	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

Bits	Name	Description	R/W
31:0	MSKS n	Writing 1 to MSKS n will set MSK n to 1 in register PDMSK. Writing 0 to MSKS n will no use.	WC

19.4.4.7 PORT D Mask Clear Registers (PDMSKC,0x28)

PORT D GROUP mask clear registers.

	PDMSKC																												0x28
Bit	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
	MSKC31	MSKC30	MSKC29	MSKC28	MSKC27	MSKC26	MSKC25	MSKC24	MSKC23	MSKC22	MSKC21	MSKC20	MSKC19	MSKC18	MSKC17	MSKC16	MSKC15	MSKC14	MSKC13	MSKC12	MSKC11	MSKC10	MSKC09	MSKC08	MSKC07	MSKC06	MSKC05	MSKC04	MSKC03	MSKC02	MSKC01	MSKC00
RST	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

Bits	Name	Description	R/W
31:0	MSKC n	Writing 1 to MSKC n will set MSK n to 0 in register PDMSK. Writing 0 to MSKC n will no use.	WC

19.4.4.8 PORT D PAT1/Direction Registers (PDPAT1,0x30)

PORT D GROUP pattern1/direction registers.

	PDPAT1																												0x30
Bit	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
	PAT131	PAT130	PAT129	PAT128	PAT127	PAT126	PAT125	PAT124	PAT123	PAT122	PAT121	PAT120	PAT119	PAT118	PAT117	PAT116	PAT115	PAT114	PAT113	PAT112	PAT111	PAT110	PAT109	PAT108	PAT107	PAT106	PAT105	PAT104	PAT103	PAT102	PAT101	PAT100
RST	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	1	1	1	1	1

Bits	Name	Description	R/W
31:0	PAT1 n	Where n = 0 ~ 31 and PAT1 n = PAT131 ~ PAT10. When INT n = 1 (Interrupt function): 0: Level trigger interrupt 1: Edge trigger interrupt When INT n = 0 and MSK = 0 (Device function): 0: Corresponding pin is used as device 0 or device 1 function 1: Corresponding pin is used as device 2 or device 3 function When INT n = 0 and MSK = 1 (GPIO function): 0: Corresponding pin is used as gpio output 1: Corresponding pin is used as gpio input	RW

19.4.4.9 PORT D PAT1/Direction Set Registers (PDPAT1S,0x34)

PORT D GROUP pattern1 or direction set registers.

	PDPAT1S																													0x34
Bit	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
	PAT1S31	PAT1S30	PAT1S29	PAT1S28	PAT1S27	PAT1S26	PAT1S25	PAT1S24	PAT1S23	PAT1S22	PAT1S21	PAT1S20	PAT1S19	PAT1S18	PAT1S17	PAT1S16	PAT1S15	PAT1S14	PAT1S13	PAT1S12	PAT1S11	PAT1S10	PAT1S09	PAT1S08	PAT1S07	PAT1S06	PAT1S05	PAT1S04	PAT1S03	PAT1S02	PAT1S01	PAT1S00
RST	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

Bits	Name	Description	R/W
31:0	PAT1S n	Writing 1 to PAT1S n will set PAT1 n to 1 in register PAPAT1. Writing 0 to PAT1S n will no use.	WC

19.4.4.10 PORT D PAT1/Direction Clear Registers (PDPAT1C,0x38)

PORT D GROUP pattern1 or direction clear registers.

	PDPAT1S,																													0x38
Bit	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
	PAT1C31	PAT1C30	PAT1C29	PAT1C28	PAT1C27	PAT1C26	PAT1C25	PAT1C24	PAT1C23	PAT1C22	PAT1C21	PAT1C20	PAT1C19	PAT1C18	PAT1C17	PAT1C16	PAT1C15	PAT1C14	PAT1C13	PAT1C12	PAT1C11	PAT1C10	PAT1C09	PAT1C08	PAT1C07	PAT1C06	PAT1C05	PAT1C04	PAT1C03	PAT1C02	PAT1C01	PAT1C00
RST	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

Bits	Name	Description	R/W
31:0	PAT1C n	Writing 1 to PAT1C n will set PAT1 n to 0 in register PAPAT1. Writing 0 to PAT1C n will no use.	WC

19.4.4.11 PORT D PAT0/Data Registers (PDPAT0,0x40)

PORT D GROUP pattern 0 or data registers.

	PDPAT0,																													0x40
Bit	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
	PAT031	PAT030	PAT029	PAT028	PAT027	PAT026	PAT025	PAT024	PAT023	PAT022	PAT021	PAT020	PAT019	PAT018	PAT017	PAT016	PAT015	PAT014	PAT013	PAT012	PAT011	PAT010	PAT009	PAT008	PAT007	PAT006	PAT005	PAT004	PAT003	PAT002	PAT001	PAT000
RST	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

Bits	Name	Description	R/W
31:0	PAT0 n	Where n = 0 ~ 31 and PAT0 n = PAT00 ~ PAT031. When INTn = 1 and PAT1 = 0: 0: Port is low level triggered interrupt input 1: Port is low high triggered interrupt input When INTn = 1 and PAT1 = 1: 0: Port is falling edge triggered interrupt input 1: Port is rising edge triggered interrupt input When INTn = 0 and MSK = 0 and PAT1 = 0: 0: Port is pin of device 0 1: Port is pin of device 1 When INTn = 0 and MSK = 0 and PAT1 = 1: 0: Port is pin of device 2 1: Port is pin of device 3 When INTn = 0 and MSK = 1 and PAT1 = 0: 0: Port is GPIO output 0 1: Port is GPIO output 1	RW

19.4.4.12 PORT D PAT0/Data Set Registers (PDPAT0S,0x44)

PORT D GROUP pattern0 or data set registers.

	PDPAT0S																													0x44
Bit	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
	PAT0S31	PAT0S30	PAT0S29	PAT0S28	PAT0S27	PAT0S26	PAT0S25	PAT0S24	PAT0S23	PAT0S22	PAT0S21	PAT0S20	PAT0S19	PAT0S18	PAT0S17	PAT0S16	PAT0S15	PAT0S14	PAT0S13	PAT0S12	PAT0S11	PAT0S10	PAT0S09	PAT0S08	PAT0S07	PAT0S06	PAT0S05	PAT0S04	PAT0S03	PAT0S02	PAT0S01	PAT0S00
RST	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

Bits	Name	Description	R/W
31:0	PAT0S n	Writing 1 to PAT0S n will set PAT0 n to 1 in register PAPAT0. Writing 0 to PAT0S n will no use.	WC

19.4.4.13 PORT D PAT0/Data Clear Registers (PDPAT0C,0x48)

PORT D GROUP pattern 0 or data clear registers.

	PDPAT0C																													0x48
Bit	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
	PAT0C31	PAT0C30	PAT0C29	PAT0C28	PAT0C27	PAT0C26	PAT0C25	PAT0C24	PAT0C23	PAT0C22	PAT0C21	PAT0C20	PAT0C19	PAT0C18	PAT0C17	PAT0C16	PAT0C15	PAT0C14	PAT0C13	PAT0C12	PAT0C11	PAT0C10	PAT0C09	PAT0C08	PAT0C07	PAT0C06	PAT0C05	PAT0C04	PAT0C03	PAT0C02	PAT0C01	PAT0C00
RST	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

Bits	Name	Description	R/W
31:0	PAT0C n	Writing 1 to PAT0C n will set PAT0 n to 0 in register PAPAT0. Writing 0 to PAT0C n will no use.	WC

19.4.4.14 PORT D FLAG Registers (PDFLG,0x50)

PORT D GROUP flag registers. They are read-only registers.

	PDFLG																													0x50
Bit	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
	FLAG31	FLAG30	FLAG29	FLAG28	FLAG27	FLAG26	FLAG25	FLAG24	FLAG23	FLAG22	FLAG21	FLAG20	FLAG19	FLAG18	FLAG17	FLAG16	FLAG15	FLAG14	FLAG13	FLAG12	FLAG11	FLAG10	FLAG09	FLAG08	FLAG07	FLAG06	FLAG05	FLAG04	FLAG03	FLAG02	FLAG01	FLAG00
RST	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

Bits	Name	Description	R/W
31:0	FLAG n	Where n = 0 ~ 31 and FLAG n = FLAG00 ~ FLAG31. FLAG n is interrupt flag bit for checking the interrupt whether to happen. When GPIO is used as interrupt function and the interrupt happened, the FLAG n in PDFLG will be set to 1.	R

NOTE:

1. Not every PAD support Input Enable Control, please check out Function Overview chapter for details.

19.4.4.15 PORT D FLAG Clear Registers (PDFLGC,0x58)

PORT D GROUP flag clear registers. They are read-only registers.

	PDFLGC																													0x58
Bit	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
	FLAGC31	FLAGC30	FLAGC29	FLAGC28	FLAGC27	FLAGC26	FLAGC25	FLAGC24	FLAGC23	FLAGC22	FLAGC21	FLAGC20	FLAGC19	FLAGC18	FLAGC17	FLAGC16	FLAGC15	FLAGC14	FLAGC13	FLAGC12	FLAGC11	FLAGC10	FLAGC09	FLAGC08	FLAGC07	FLAGC06	FLAGC05	FLAGC04	FLAGC03	FLAGC02	FLAGC01	FLAGC00
RST	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

Bits	Name	Description	R/W
31:0	FLAGC n	When GPIO is used as interrupt function and when write 1 to the bit, the bit FLAG n in PDFLG will be cleared.	RC

19.4.4.16 PORT D PULL Disable Registers (PDPEN,0x70)

PORT D GROUP pull disable registers.

	PDPEN																													0x70
Bit	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
	PEN31	PEN30	PEN29	PEN28	PEN27	PEN26	PEN25	PEN24	PEN23	PEN22	PEN21	PEN20	PEN19	PEN18	PEN17	PEN16	PEN15	PEN14	PEN13	PEN12	PEN11	PEN10	PEN09	PEN08	PEN07	PEN06	PEN05	PEN04	PEN03	PEN02	PEN01	PEN00
RST	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

Bits	Name	Description	R/W
31:0	PEN n	Where n = 0 ~ 31 and PEN n = PEN0 ~ PEN31. PEN n is used for setting the port to be PULL function enable. 0: port PULL enable 1: port PULL disable	RW

19.4.4.17 PORT D PULL Set Registers (PDPENS,0x74)

PORT D GROUP pull set registers. They are write-only registers.

	PDPENS																													0x74
Bit	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
	PENS31	PENS30	PENS29	PENS28	PENS27	PENS26	PENS25	PENS24	PENS23	PENS22	PENS21	PENS20	PENS19	PENS18	PENS17	PENS16	PENS15	PENS14	PENS13	PENS12	PENS11	PENS10	PENS09	PENS08	PENS07	PENS06	PENS05	PENS04	PENS03	PENS02	PENS01	PENS00
RST	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

Bits	Name	Description	R/W
31:0	PENS n	Writing 1 to PENS n will set PEN n to 1 in register PDPEN. Writing 0 to PENS n will no use.	WC

19.4.4.18 PORT D PULL Clear Registers (PDPENC,0x78)

PORT pull clear registers. They are write-only registers.

	PDPENC																													0x78
Bit	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
	PENC31	PENC30	PENC29	PENC28	PENC27	PENC26	PENC25	PENC24	PENC23	PENC22	PENC21	PENC20	PENC19	PENC18	PENC17	PENC16	PENC15	PENC14	PENC13	PENC12	PENC11	PENC10	PENC09	PENC08	PENC07	PENC06	PENC05	PENC04	PENC03	PENC02	PENC01	PENC00
RST	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

Bits	Name	Description	R/W
31:0	PENC n	Writing 1 to PENC n will set PEN n to 0 in register PDPEN. Writing 0 to PENC n will no use.	WC

19.4.5 PORT Z Shadow Register Group

PORT Z group only have eight register, PZINTS, PZINTC, PZMSKS, PZMSKC, PZPAT1S, PZPAT1C, PZPAT0S, PZPAT0C. those registers shadow as same as other group registers function which the offset address equal.

	PZINTS																													0x14
Bit	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
	INTS31	INTS30	INTS29	INTS28	INTS27	INTS26	INTS25	INTS24	INTS23	INTS22	INTS21	INTS20	INTS19	INTS18	INTS17	INTS16	INTS15	INTS14	INTS13	INTS12	INTS11	INTS10	INTS09	INTS08	INTS07	INTS06	INTS05	INTS04	INTS03	INTS02	INTS01	INTS00
RST	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

	PZINTC																													0x18
Bit	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
	INTC31	INTC30	INTC29	INTC28	INTC27	INTC26	INTC25	INTC24	INTC23	INTC22	INTC21	INTC20	INTC19	INTC18	INTC17	INTC16	INTC15	INTC14	INTC13	INTC12	INTC11	INTC10	INTC09	INTC08	INTC07	INTC06	INTC05	INTC04	INTC03	INTC02	INTC01	INTC00
RST	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

	PZMSKS																													0x24
Bit	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
	MSKS31	MSKS30	MSKS29	MSKS28	MSKS27	MSKS26	MSKS25	MSKS24	MSKS23	MSKS22	MSKS21	MSKS20	MSKS19	MSKS18	MSKS17	MSKS16	MSKS15	MSKS14	MSKS13	MSKS12	MSKS11	MSKS10	MSKS09	MSKS08	MSKS07	MSKS06	MSKS05	MSKS04	MSKS03	MSKS02	MSKS01	MSKS00
RST	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

	PZMSKC																													0x28
Bit	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
	MSKC31	MSKC30	MSKC29	MSKC28	MSKC27	MSKC26	MSKC25	MSKC24	MSKC23	MSKC22	MSKC21	MSKC20	MSKC19	MSKC18	MSKC17	MSKC16	MSKC15	MSKC14	MSKC13	MSKC12	MSKC11	MSKC10	MSKC09	MSKC08	MSKC07	MSKC06	MSKC05	MSKC04	MSKC03	MSKC02	MSKC01	MSKC00
RST	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

	PZPAT1S																													0x34
Bit	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
	PAT1S31	PAT1S30	PAT1S29	PAT1S28	PAT1S27	PAT1S26	PAT1S25	PAT1S24	PAT1S23	PAT1S22	PAT1S21	PAT1S20	PAT1S19	PAT1S18	PAT1S17	PAT1S16	PAT1S15	PAT1S14	PAT1S13	PAT1S12	PAT1S11	PAT1S10	PAT1S09	PAT1S08	PAT1S07	PAT1S06	PAT1S05	PAT1S04	PAT1S03	PAT1S02	PAT1S01	PAT1S00
RST	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

	PZPAT1S,																													0x38
Bit	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
	PAT1C31	PAT1C30	PAT1C29	PAT1C28	PAT1C27	PAT1C26	PAT1C25	PAT1C24	PAT1C23	PAT1C22	PAT1C21	PAT1C20	PAT1C19	PAT1C18	PAT1C17	PAT1C16	PAT1C15	PAT1C14	PAT1C13	PAT1C12	PAT1C11	PAT1C10	PAT1C09	PAT1C08	PAT1C07	PAT1C06	PAT1C05	PAT1C04	PAT1C03	PAT1C02	PAT1C01	PAT1C00
RST	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

	PZPAT0S																													0x44
Bit	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
	PAT0S31	PAT0S30	PAT0S29	PAT0S28	PAT0S27	PAT0S26	PAT0S25	PAT0S24	PAT0S23	PAT0S22	PAT0S21	PAT0S20	PAT0S19	PAT0S18	PAT0S17	PAT0S16	PAT0S15	PAT0S14	PAT0S13	PAT0S12	PAT0S11	PAT0S10	PAT0S09	PAT0S08	PAT0S07	PAT0S06	PAT0S05	PAT0S04	PAT0S03	PAT0S02	PAT0S01	PAT0S00
RST	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

	PZPAT0C																														0x48
Bit	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
	PAT0C31	PAT0C30	PAT0C29	PAT0C28	PAT0C27	PAT0C26	PAT0C25	PAT0C24	PAT0C23	PAT0C22	PAT0C21	PAT0C20	PAT0C19	PAT0C18	PAT0C17	PAT0C16	PAT0C15	PAT0C14	PAT0C13	PAT0C12	PAT0C11	PAT0C10	PAT0C09	PAT0C08	PAT0C07	PAT0C06	PAT0C05	PAT0C04	PAT0C03	PAT0C02	PAT0C01	PAT0C00
RST	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

19.4.6 GPIOZ Group ID to Load Register (PzGID2LD,0xF0)

This register controls which Port Group to be load from GPIOZ.

	PzGID2LD																														0xF0
Bit	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
	Reserved																													GPID		
RST	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

GPIOZ Group have four shadow register for GPIO Group A to D, when set or clear A to D group register below four type, it will through shadow register to implement. the GPIOZ ID used for controls which Group to be manipulate.

Bits	Name	Description	RW
31:3	Reserved	Writing has no effect Read as zero	R
2:0	GPID	GPIO Port ID load control register, according the values of GPID to load Shadow register group to the same offset address of group ID meet the GPID.	W

GPID	Shadow register load to work groups
0	Load to GPIO Port A
1	Load to GPIO Port B
2	Load to GPIO Port C
3	Load to GPIO Port D
4	Reserved
5	Reserved

19.5 Program Guide

19.5.1 Port Function Guide

INT	MASK	PAT1	PAT0	Port Description
1	0	0	0	Port is low level triggered interrupt input.
1	0	0	1	Port is high level triggered interrupt input.
1	0	1	0	Port is fall edge triggered interrupt input.
1	0	1	1	Port is rise edge triggered interrupt input.
1	1	0	0	Port is low level triggered interrupt input. Interrupt is masked. Flag is recorded.
1	1	0	1	Port is high level triggered interrupt input. Interrupt is masked. Flag is recorded.
1	1	1	0	Port is fall edge triggered interrupt input. Interrupt is masked. Flag is recorded.
1	1	1	1	Port is rise edge triggered interrupt input. Interrupt is masked. Flag is recorded.
0	0	0	0	Port is pin of device 0.
0	0	0	1	Port is pin of device 1.
0	0	1	0	Port is pin of device 2.
0	0	1	1	Port is pin of device 3.
0	1	0	0	Port is GPIO output 0.
0	1	0	1	Port is GPIO output 1.
0	1	1	?	Port is GPIO input.

19.5.2 Configure without 3rd-unexpected state

Function switching between devices and gpio is controlled by combination of 4 registers: INT/MASK/PAT1/PAT0. As the four register is accessed by soft ware in serial sequence, which posing a risk that third states occur during configuration.

For example:

current state is:

INT = 0x0, MASK = 0x0, PAT0 = 0x0, PAT1 = 0x0;

target state is:

INT = 0x1, MASK = 0x1, PAT0 = 0x1, PAT1 = 0x1;

If the first step is to configure INT to 0x1, the 3rd state is:

INT = 0x1, MASK = 0x0, PAT0 = 0x0, PAT1 = 0x0;

This might triggers interrupt if port is low level.

To avoid the unexpected 3-state, the four registers must be updated in the same time.

There is an dummy group GPIOZ, which only used in this purpose. The working mechanism is the information of configure can be temporarily stored in the GPIOZ. When assert GPIOZ.LD, all the four registers will load their content into target group at the same time.

Example:

If the Port B current state is:

INT = 0x0, MASK = 0x0, PAT0 = 0x0, PAT1 = 0x0;

It needs to be configured to state:

INT = 0x1, MASK = 0x1, PAT0 = 0x1, PAT1 = 0x1;

There are two steps to finish:

Step1: put the configuration to temporary buffer: GPIOZ

PzINTS = 0x1;

PzMSKS = 0x1;

PzPAT0S = 0x1;

PzPAT1S = 0x1;

alternatively, PzINT, PzMSK, PzPAT0, PzPAT1 are also can be configured straightly.

Step2: configure PzGID2LD to specify which port group to load

PzGID2LD = 0x1; (0x1 stands for GPIO Port B)

20 SMB Controller

20.1 Overview

The SMB bus is a two-wire serial interface, consisting of a serial data line (SDA) and a serial clock (SCL). These wires carry information between the devices connected to the bus. Each device is recognized by a unique address and can operate as either a “transmitter” or “receiver,” depending on the function of the device. Devices can also be considered as masters or slaves when performing data transfers. A master is the device that initializes/terminates a data transfer on the bus and generates clock signals to permit that transfer. During that time, any addressed device is considered as a slave. The SMB controller is software controlled. It behaves as a master or a slave. However, operating as a master and slave simultaneously is not supported.

20.1.1 Features

- Two-wire SMB serial interface – consists of a serial data line (SDA) and a serial clock (SCL)
- Two speeds
  - Standard mode (100 Kb/s)
  - Fast mode (400 Kb/s)
- Device clock is identical with pclk(APB clock)
- Programmable SCL generator
- Master or slave SMB operation
- 7-bit addressing/10-bit addressing whether configure as Master or Slave
- Transmit General Call or START byte
- level transmit and receive 64 depth FIFOs
- Interrupt operation
- The number of devices that you can connect to the same SMB-bus is limited only by the maximum bus capacitance of 400pF
- APB interface
- 3 independent SMB channels (SMB0, SMB1, SMB2)

20.1.2 Pin Description

Table 20-1 SMB Pin Description

Name	Width	GPIO	Description
SDA0	1-bit	PB24	SMB serial data
SCL0	1-bit	PB23	SMB serial clock
SDA1	1-bit	PA01/ PC27	SMB serial data
SCL1	1-bit	PA00/ PC26	SMB serial clock
SDA2	1-bit	PD01	SMB serial data
SCL2	1-bit	PD00	SMB serial clock

Note:
1.Dedicate pull-up resistors must be introduced on board-level. The low-to-high (rise time) transition is highly dependent on RC time constant of the bus.
2.The two GPIOs refer to SMB must belong to the same GPIO group.

Totally speaking, for standard-mode SMB-bus system, the pull up resistor depends on following parameters:

- Supply voltage
- Bus capacitance
- Number of connected devices

For fast-mode SMB-bus system, switched pull-up circuit may be essential for strict speed and load requirement. (Please refer NXP Semiconductors, *UM10204: I2C-Bus specification and user manual (Rev.6--4 April 2014)*')

20.2 Registers

20.2.1 Registers Memory Map

A read operation to an address location that contains unused bits results in a 0 value being returned on each of the unused bits.

Registers in SMB controller can be accessed by indicating 24-bit Address Base combined with 8-bit Address Offset.

Table 20-2 Registers Memory Map-Address Base

Name	Addr Base	Description
SMB0	0x10050000	Address base of SMB0
SMB1	0x10051000	Address base of SMB1
SMB2	0x10052000	Address base of SMB2

Table 20-3 Registers Memory Map-Address Offset

Name	Addr Offset	Description	Width	RW	Reset
SMB_CON	0x00	SMB control	8bits	RW	0x7D
SMB_TAR	0x04	SMB target address	13bits	RW	0x1055
SMB_SAR	0x08	SMB slave address	10bits	RW	0x055
SMB_DC	0x10	SMB data buffer and command	9bits	RW	0x000
SMB_SHCNT	0x14	Standard speed SMB SCL high count	16bits	RW	0x0190
SMB_SLCNT	0x18	Standard speed SMB SCL low count	16bits	RW	0x01D6
SMB_FHCNT	0x1C	Fast speed SMB SCL high count	16bits	RW	0x003C
SMB_FLCNT	0x20	Fast speed SMB SCL low count	16bits	RW	0x0082
SMB_INTST	0x2C	SMB Interrupt Status	12bits	R	0x000

SMB_INTM	0x30	SMB Interrupt Mask	12bits	R/W	12'h8FF
SMB_RINTST	0x34	SMB Raw Interrupt Status	12bits	R	0x0
SMB_RXTL	0x38	SMB RxFIFO Threshold	6 bits	RW	0x1F
SMB_TXTL	0x3C	SMB TxFIFO Threshold	6 bits	RW	0x20
SMB_CINT	0x40	Clear Interrupts	1 bit	R	0x0
SMB_CRXUF	0x44	Clear RXUF Interrupt	1 bit	R	0x0
SMB_CRXOF	0x48	Clear RX_OVER Interrupt	1 bit	R	0x0
SMB_CTXOF	0x4C	Clear TX_OVER Interrupt	1 bit	R	0x0
SMB_CRXREQ	0x50	Clear RDREQ Interrupt	1 bit	R	0x0
SMB_CTXABT	0x54	Clear TX_ABRT Interrupt	1 bit	R	0x0
SMB_CRXDN	0x58	Clear RX_DONE Interrupt	1 bit	R	0x0
SMB_CACT	0x5c	Clear ACTIVITY Interrupt	1 bit	R	0x0
SMB_CSTP	0x60	Clear STOP Interrupt	1 bit	R	0x0
SMB_CSTT	0x64	Clear START Interrupt	1 bit	R	0x0
SMB_CGC	0x68	Clear GEN_CALL Interrupt	1 bit	R	0x0
SMB_ENABLE	0x6C	SMB Enable	1 bit	RW	0x0
SMB_ST	0x70	SMB Status register	7 bits	R	0x06
SMB_TXFLR	0x74	TxFIFO Level Register	6 bits	R	0x0
SMB_RXFLR	0X78	RxFIFO Level Register	6 bits	R	0x0
SMB_SDAHD	0x7C	SMB SDA Hold time Register	16 bits	RW	0x55
SMB_ABTSRC	0x80	SMB Transmit Abort Status Register	32 bits	R	0x0000
SMB_DMACR	0x88	DMA Control Register	2 bits	R/W	0x0
SMB_DMATDLR	0x8c	DMA Transmit Data Level	6 bits	R/W	0x0
SMB_DMARDLR	0x90	DMA Receive Data Level	6 bits	R/W	0x0
SMB_SDASU	0x94	SMB SDA Setup Register	8 bits	RW	0x64
SMB_ACKGC	0x98	SMB ACK General Call Register	1 bit	RW	0x1
SMB_ENBST	0x9C	SMB Enable Status Register	3 bits	R	0x0
SMB_FLT	0xA0	SMB Filter Register	8bits	RW	0x1

20.2.2 Registers and Fields Description

20.2.2.1 SMB_CON (SMB Control Register)

This register can be written only when SMB is disabled, which corresponds to SMB_ENABLE[0] being set to 0. Writes at other time have on effect.

	SMB_CON																BASE + 0x00															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	Reserved									SLVDIS	RESTART	MATP	SATP	SPEED		MD
RST																	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1

Bits	Name	Description	RW
15:7	Reserved	Writing has no effect, read as zero.	R
6	SLVDIS	This bit controls whether SMB has its slave disabled. 0: slave is enabled 1: slave is disabled Note: Software should ensure that if this bit is written with'0', then bit 0 should also be written with '0'.	RW
5	RESTART	Determines whether RESTART conditions may be sent when acting as a master. Some older slaves do not support handling RESTART conditions. 0: disable 1: enable	RW
4	MATP	This bit is a read-only copy of SMB_TAR.MATP. 0: 7-bit addressing 1: 10-bit addressing	R
3	SATP	When acting as a slave, this bit controls whether the SMB responds to 7-bit or 10-bit addresses. 0: 7-bit addressing 1: 10-bit addressing	RW
2:1	SPEED	These bits control at which speed the SMB operates. 1: standard mode (100 kbps) 2: fast mode (400 kbps) **NOTE:** when these two bits are set to 2'b00 or 2'b11, the speed mode will be automatically set to 2'b10 i.e. fast mode.	RW
0	MD	This bit controls whether the SMB master is enabled. 0: master disabled 1: master enabled Note: Software should ensure that if this bit is written with'1', then bit 6 should also be written with '1'.	RW

20.2.2.2 SMB_TAR (SMB Target Address Register)

	SMB_TAR																BASE + 0x04													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	Reserved			MATP	SPECIAL	GC_OR_START	SMBTAR								
RST																	0	0	0	1	0	0	0	0	0	1	0	1	0	1	0	1

Bits	Name	Description	R/W
15:13	Reserved	Writing has no effect, read as zero.	R
12	MATP	This bit controls whether the SMB starts its transfers in 7- or 10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing **NOTE:** this bit is initially set to 0.	RW
11	SPECIAL	This bit indicates whether software performs a General Call or START BYTE command. 0: ignore the bit of GC_OR_START and use SMBTAR normally 1: perform special SMB command as specified in GC_OR_START bit **NOTE:** this bit is initially set to 0	RW
10	GC_OR_START	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a General Call or START byte command is to be performed by the SMB. 0: General Call Address – after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the SMB_INTST register. The SMB remains in General Call mode until the SPECIAL bit value (bit 11) is cleared 1: START BYTE	RW
9:0	SMBTAR	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits. If the SMB_TAR and SMB_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself, but only to a slave.	RW

NOTE:

(1).It is not necessary to perform any write to this register if SMB is enabled as an SMB slave only.

(2).The meaning of "START byte" in bit 10 is different with the SMB "start condition".

"START byte" can only be applied when a device does not have such an interface, it must constantly monitor the bus via software. For more detail information, please refer *UM10204: I2C-Bus specification and user manual (Rev.6--4 April 2014), '*3.1.15 START byte' section.

"start condition" means A HIGH to LOW transition on the SDA line while SCL is HIGH defines a START condition. please refer *UM10204: I2C-Bus specification and user manual (Rev.6--4 April 2014), '*3.1.4 START and STOP' section for more information.

20.2.2.3 SMB_SAR (SMB Slave Address Register)

	SMB_SAR																BASE + 0x08													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

																Reserved						SMBSAR									
RST																0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1

Bits	Name	Description	R/W
15:10	Reserved	Writing has no effect, read as zero.	R
9:0	SMBSAR	The SMBSAR holds the slave address when the SMB is operating as a slave. For 7-bit addressing, only SMBSAR[6:0] is used. This register can be written only when the SMB interface is disabled. Writes at other times have no effect.	RW

NOTE: It is not necessary to perform any write to this register if SMB is enabled as an SMB master only.

20.2.2.4 SMB_DC (SMB Rx/Tx Data Buffer and Command Register)

This is the register the CPU writes to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO.

	SMB_DC																BASE + 0x10														
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	Reserved					RESTART	STOP	CMD	DAT							
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:11	Reserved	Writing has no effect, read as zero.	R
10	RESTART	This bit controls whether a RESTART is issued before the byte is sent or received. 1: If SMB_CON.RESTART is 1, a RESTART is issued before the byte is sent or received, regardless of whether or not the transfer direction is changing from the previous command; if SMB_CON.RESTART is 0, a STOP followed by a START is issued instead. 0: If SMB_CON.RESTART is 1, a RESTERT is issued if the transfer direction is changing from the previous command, if SMB_CON.RESTART is 0, a STOP followed by a START is issued instead.	W
9	STOP	This bit controls whether a STOP is issued after the byte is sent or received.	W

		1: STOP is issued after the byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master immediately tries to start a new transfer by issuing a START and arbitrating for the bus. 0: STOP is not issued after the byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master continues the current transfer by sending/receiving data bytes according to the value of the CMD bit. If the Tx FIFO is empty, the master holds the SCL line low and stalls the bus until a new command is available in the Tx FIFO.	
8	CMD	This bit controls whether a read or a write is performed. This bit does not control the direction when the SMB acts as a slave. It controls only the direction when it acts as a master. 1: Read 0: Write When programming this bit, you should remember the following: attempting to perform a read operation after a General Call command has been sent results in a *TX_ABRT* interrupt (bit 6 of the *SMB_RAW_INTR_STAT* register), unless bit 11 (*SPECIAL*) in the *SMB_TAR* register has been cleared. If a "1" is written to this bit after receiving a *RD_REQ* interrupt, then a *TX_ABRT* interrupt occurs.	W
7:0	DAT	This register contains the data to be transmitted or received on the SMB bus. If you are writing to this register and want to perform a read, bits 7:0 (*DAT*) are ignored by the SMB module. However, when you read this register, these bits return the value of data received on the SMB interface.	RW

NOTE: [1].this command only transfer 8-bit data combined with 1-bit CMD, extra bits on the bus will be eliminated. i.e. only 8-0 bits on the bus are accepted by SMB controller.
[2].For the detail information about what condition could generate START, STOP and RESTART, please refer "*1.3.6 The condition could generate START, STOP and RESTART*".
[3].The differences between *SMB_ENABLE[5]*(RESTART) and *SMB_DC[10]*(RESTART) is that SMB_ENABLE[5] determine whether or not SMB the capability of generating repeat start, but not transmit RESTART when this bit be setted. If this bit set "1", SMB_DC[10] controls whether a RESTART is issued, if set "0", SMB_DC[10] have no meaning.

20.2.2.5 SMB_SHCNT (SMB Standard Speed SCL High Count Register)

	SMB_SHCNT																BASE + 0x14															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	SMBSHCNT															
RST																	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0

Bits	Name	Description	R/W
15:0	SMBSHCNT	This register must be set before any SMB bus transaction can take place to ensure proper I/O timing. The register sets the SCL clock high-period count for standard speed. This register can be written only when the SMB interface is disabled which corresponds to SMB_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set.	RW

20.2.2.6 SMB_SLCNT (SMB Standard Speed SCL Low Count Register)

	SMB_SLCNT															BASE + 0x18													
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMBSLCNT															
RST																0	0	0	0	0	0	0	1	1	1	0	1	0	1	1	0

Bits	Name	Description	R/W
15:0	SMBSLCNT	This register must be set before any SMB bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed. This register can be written only when the SMB interface is disabled which corresponds to SMB_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 8; hardware prevents values less than this being written, and if attempted, results in 8 being set.	RW

20.2.2.7 SMB_FHCNT (SMB Fast Speed SCL High Count Register)

	SMB_FHCNT															BASE + 0x1C													
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																SMBFHCNT															
RST																0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

Bits	Name	Description	R/W
15:0	SMBFHCNT	This register must be set before any SMB bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast speed.	RW

		This register can be written only when the SMB interface is disabled which corresponds to SMB_ENABLE[0] being set to 0. Writes at other times have no effect. Minimum value allowed for the SMBSHCNT registers is 6. If the set value was less than 6, it will be automatically set to 6.	

20.2.2.8 SMB_FLCNT (SMB Fast Speed SCL Low Count Register)

	SMB_FLCNT																BASE + 0x20													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	SMBFLCNT															
RST																	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0

Bits	Name	Description	R/W
15:0	SMBFLCNT	This register must be set before any SMB bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast speed. This register can be written only when the SMB interface is disabled which corresponds to SMB_ENABLE[0] being set to 0. Writes at other times have no effect. Minimum value that can be programmed in the SMBSLCNT registers is 8. If the set value was less than 8, it will be automatically set to 8	RW

20.2.2.9 SMB_INTST (SMB Interrupt Status Register)

Each bit in this register has a corresponding mask bit in the SMBINTM register. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in the *SMB_RAW_INTR_STAT* register

	SMB_INTST																BASE + 0x2C													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	Reserved				IGC	ISTT	ISTP	IACT	RXDN	TXABT	RDREQ	TXEMP	TXOF	RXFL	RXOF	RXUF
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:12	Reserved	Writing has no effect, read as zero.	R
11	IGC	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling SMB or when the CPU	R

		reads bit 0 of the *SMB_CGC* register. SMB stores the received data in the Rx buffer.	
10	ISTT	Indicates whether a START or RESTART condition has occurred on the SMB interface regardless of whether SMB is operating in slave or master mode.	R
9	ISTP	Indicates whether a STOP condition has occurred on the SMB interface regardless of whether SMB is operating in slave or master mode.	R
8	IACT	This bit captures SMB activity and stays set until it is cleared. There are four ways to clear it: 1 Disabling the SMB 2 Reading the SMB_CACT register 3 Reading the SMB_CINT register 4 System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the SMB module is idle, this bit remains set until cleared, indicating that there was activity on the bus.	R
7	RXDN	When the SMB is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done.	R
6	TXABT	This bit indicates if SMB, as an SMB transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an SMB master or an SMB slave, and is referred to as a “transmit abort”. When this bit is set to 1, the SMB_ABTSRC register indicates the reason why the transmit abort takes places. **NOTE:** The SMB flushes/resets/empties the TX FIFO whenever this bit is set. The TX FIFO remains in this flushed state until the register SMB_CTXABT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface.	R
5	RDREQ	This bit is set to 1 when SMB is acting as a slave and another SMB master is attempting to read data from SMB. The SMB holds the SMB bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the SMB_DC register. This bit is set to 0 just after the processor reads the SMB_CRREQ register.	R
4	TXEMP	This bit is set to 1 when the transmit buffer is at or below the threshold value set in the SMB_TXTL register. It is automatically cleared by hardware when the buffer level goes above the threshold. When the SMB_ENB bit 0 is 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer activity, then with SMB_ENB[0]=0.	R
3	TXOF	Set during transmit if the transmit buffer is filled to	R

		SMB_TX_BUFFER_DEPTH and the processor attempts to issue another SMB command by writing to the SMB_DC register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when SMB_ENB[0]=0, this interrupt is cleared.	
2	RXFL	Set when the receive buffer reaches or goes above the SMB_RXTL threshold in the SMB_RXTL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (SMB_ENB[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the SMB_ENB bit 0 is set to 0, regardless of the activity that continues.	R
1	RXOF	Set if the receive buffer is completely filled to 64 and an additional byte is received from an external SMB device. The SMB acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (SMB_ENB[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when SMB_ENB[0]=0, this interrupt is cleared.	R
0	RXUF	Set if the processor attempts to read the receive buffer when it is empty by reading from the SMB_DC register. If the module is disabled (SMB_ENB[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when SMB_ENB[0]=0, this interrupt is cleared.	R

NOTE: *1.3.5* summary all the condition could flush TX FIFO.

20.2.2.10 SMB_INTM (SMB Interrupt Mask Register)

These bits mask their corresponding interrupt status bits. This register is active low; a value of 0 masks the interrupt, whereas a value of 1 unmasks the interrupt.

	SMB_INTM																BASE + 0x30											
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	Reserved				MIGC	MISTT	MISTP	MIACT	MRXDN	MTXABT	MRDREQ	MTXEMP	MTXOF	MRXFL	MRXOF	MRXUF
RST																	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1

Bits	Name	Description	R/W
15:12	Reserved	Writing has no effect, read as zero.	R
11	MIGC	These bits mask their corresponding interrupt status bits in the SMBINTST register.	RW
10	MISTT		RW
9	MISTP		RW
8	MIACT		RW
7	MRXDN		RW

6	MTXABT		RW
5	MRDREQ		RW
4	MTXEMP		RW
3	MTXOF		RW
2	MRXFL		RW
1	MRXOF		RW
0	MRXUF		RW

20.2.2.11 SMB_RAW_INTR_STAT

Address: **BASE + 0x34**

Unlike the *SMB_INTST* register, these bits are not masked so they always show the true status of the SMB. See *SMB_INTST* for a detailed description of these bits. All bits are only read.

20.2.2.12 SMB_RXTL (SMB Receive FIFO Threshold Register)

	SMB_RXTL																BASE + 0x38													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										RXTL					
RST																	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Bits	Name	Description	R/W
15:6	Reserved	Writing has no effect, read as zero.	R
5:0	RXTL	Receive FIFO Threshold Level. Controls the level of entries that triggers the RxFIFO full interrupt. A value of n sets the threshold for (n+1) entries. For example, a value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries.	RW

20.2.2.13 SMB_TXTL (SMB Transmit FIFO Threshold Register)

	SMB_TXTL																BASE + 0x3C													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																										TXTL					
RST																	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	Name	Description	R/W
15:6	Reserved	Writing has no effect, read as zero.	R
5:0	TXTL	Transmit FIFO Threshold Level. Controls the level of entries that trigger the TxFIFO empty interrupt. A value of n sets the threshold for n entries. For example, A value of 0 sets the threshold for 0 entries, and a value of 255 sets the threshold for 255 entries.	RW

20.2.2.14 SMB_CINT (SMB Clear Combined and Individual Interrupt Register)

	SMB_CINT																BASE + 0x40													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																Reserved															CINT
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:1	Reserved	Writing has no effect, read as zero.	R
0	CINT	Read this register to clear the combined interrupt, all individual interrupts, and the SMB_ABTSRC register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the SMB_ABTSRC register for an exception to clearing SMB_ABTSRC.	R

20.2.2.15 SMB_CRXUF (SMB Clear RXUF Interrupt Register)

	SMB_CRXUF																BASE + 0x44													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																Reserved															CRXUF
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:1	Reserved	Writing has no effect, read as zero.	R
0	CRXUF	Read this register to clear the RXUF interrupt (bit 0) of the SMB_INTST register.	R

20.2.2.16 SMB_CRXOF (SMB Clear RXOF Interrupt Register)

	SMB_CRXOF															BASE + 0x48															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														CRXOF
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:1	Reserved	Writing has no effect, read as zero.	R
0	CRXOF	Read this register to clear the RXOF interrupt (bit 1) of the SMBINTST register.	R

20.2.2.17 SMB_CTXOF (SMB Clear TX_OVER Interrupt Register)

	SMB_CTXOF															BASE + 0x4C															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														CTXOF
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:1	Reserved	Writing has no effect, read as zero.	R
0	CTXOF	Read this register to clear the TX_OVER interrupt (bit 3) of the SMBINTST register.	R

20.2.2.18 SMB_CRXREQ (SMB Clear RDREQ Interrupt Register)

	SMB_CRXREQ															BASE + 0x50															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														CLRDREQ
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:1	Reserved	Writing has no effect, read as zero.	R
0	CLRDREQ	Read this register to clear the RDREQ interrupt (bit 5) of the SMBINTST register.	R

20.2.2.19 SMB_CTXABT (SMB Clear TX_ABRT Interrupt Register)

	SMB_CTXABT																BASE + 0x54															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															CTXABT
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:1	Reserved	Writing has no effect, read as zero.	R
0	CTXABT	Read this register to clear the TX_ABRT interrupt (bit 6) of the SMB_INTST register, and the SMB_ABTSRC register. This also releases the TX FIFO from the flushed/reset state, allowing more writes to the TX FIFO. Refer to Bit 9 of the SMB_ABTSRC register for an exception to clearing SMB_ABTSRC.	R

20.2.2.20 SMB_CRXDN (SMB Clear RX_DONE Interrupt Register)

	SMB_CRXDN																BASE + 0x58															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															CRXDN
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:1	Reserved	Writing has no effect, read as zero.	R
0	CRXDN	Read this register to clear the RX_DONE interrupt (bit 7) of the SMB_INTST register.	R

20.2.2.21 SMB_CACT (SMB Clear ACTIVITY Interrupt Register)

	SMB_CACT															BASE + 0x5C														
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																Reserved															CACT
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:1	Reserved	Writing has no effect, read as zero.	R
0	CACT	Reading this register clears the ACTIVITY interrupt if the SMB is not active anymore. If the SMB module is still active on the bus, the ACTIVITY interrupt bit continues to be set. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY interrupt (bit 8) of the SMBINTST register.	R

20.2.2.22 SMB_CSTP (SMB Clear STOP Interrupt Register)

	SMB_CSTP															BASE + 0x60														
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																Reserved															CSTP
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:1	Reserved	Writing has no effect, read as zero.	R
0	CSTP	Read this register to clear the STOP interrupt (bit 9) of the SMBINTST register.	R

20.2.2.23 SMB_CSTT (SMB Clear START Interrupt Register)

	SMB_CSTT															BASE + 0x64														
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																Reserved															CSTT
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:1	Reserved	Writing has no effect, read as zero.	R
0	CSTT	Read this register to clear the START interrupt (bit 10) of the SMBINTST register.	R

20.2.2.24 SMB_CGC (SMB Clear GEN_CALL Interrupt Register)

	SMB_CGC																BASE + 0x68															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	Reserved															CGC
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:1	Reserved	Writing has no effect, read as zero.	R
0	CGC	Read this register to clear the GEN_CALL interrupt (bit 11) of SMBINTST register.	R

20.2.2.25 SMB_ENABLE (SMB Enable Register)

	SMBE_NABLE																BASE + 0x6C															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	Reserved														ABORT	SMBENB
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:2	Reserved	Writing has no effect, read as zero.	R
1	ABORT	When set, the controller initiates the transfer abort. 0: ABORT not initiated or ABORT done 1: ABORT operation in progress The software can abort the SMB transfer in master mode by setting this bit. The software can set this bit only when ENABLE is already set; otherwise, the controller ignores any write to ABORT bit. The software cannot clear the ABORT bit once set. In response to an ABORT, the controller issues a STOP and flushes the Tx FIFO after completing the	R/W

		current transfer, then sets the TX_ABORT interrupt after the abort operation. The ABORT bit is cleared automatically after the abort operation.	
0	SMBENB	Controls whether the SMB is enabled. 0: Disables SMB (TX and RX FIFOs are held in an erased state) 1: Enables SMB Software can disable SMB while it is active. However, it is important that care be taken to ensure that SMB is disabled properly. When SMB is disabled, the following occurs: – The TX FIFO and RX FIFO get flushed. – Status bits in the SMB_INTST register are still active until SMB goes into IDLE state. If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the SMB stops the current transfer at the end of the current byte and does not acknowledge the transfer.	R/W

20.2.2.26 SMB_ST(SMB Status Register)

This is a read-only register used to indicate the current transfer status and FIFO status. The status register may be read at any time. None of the bits in this register request an interrupt.

When the SMB is disabled by writing 0 in bit 0 of the SMB_ENABLE register:

- Bits 1 and 2 are set to 1
- Bits 3 and 4 are set to 0

When the master or slave state machine goes to idle and SMB_ENABLE[0]=0:

- Bits 5 and 6 are set to 0

	SMB_ST																BASE + 0x70							
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	Reserved									SLVACT	MSTACT	RFF	RFNE	TFE	TFNF	ACT
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bits	Name	Description	R/W
15:7	Reserved	Writing has no effect, read as zero.	R
6	SLVACT	Slave FSM Activity Status. 0: Slave FSM is in IDLE state so the Slave part of SMB is not Active 1: Slave FSM is not in IDLE state so the Slave part of SMB is Active	R
5	MSTACT	Master FSM Activity Status. 0: Master FSM is in IDLE state so the Master part of SMB is not Active	R

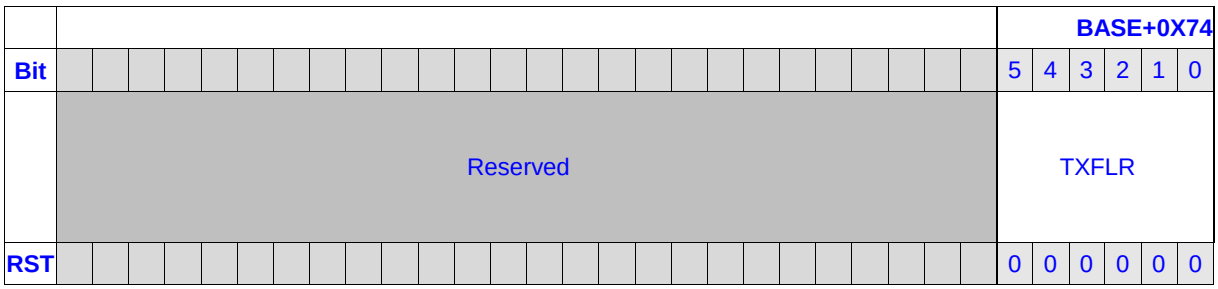
		1: Master FSM is not in IDLE state so the Master part of SMB is Active	
4	RFF	Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0: Receive FIFO is not full 1: Receive FIFO is full	R
3	RFNE	Receive FIFO Not Empty. This bit is set when the receive FIFO contains one or more entries; it is cleared when the receive FIFO is empty. 0: Receive FIFO is empty 1: Receive FIFO is not empty	R
2	TFE	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0: Transmit FIFO is not empty 1: Transmit FIFO is empty	R
1	TFNF	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. 0: Transmit FIFO is full 1: Transmit FIFO is not full	R
0	ACT	SMB Activity Status. The OR of SLVACT and MSTACT bits.	R

20.2.2.27 SMB_TXFLR(SMB Transmit FIFO Level Register)

This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:

- The SMB is disabled
- There is a transmit abort—that is, TX_ABRT bit is set in the SMB_RAW_INTR_STAT register
- The slave bulk transmit mode is aborted

The register increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.



Bits	Name	Description	R/W
31:6	Reserved	Reserved.	N/A
5:0	TXFLR	Contains the number of valid data entries in the transmit FIFO	R

20.2.2.28 SMB_RXFLR(SMB Receive FIFO Level Register)

This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:

- The SMB is disabled
- Whenever there is a transmit abort caused by any of the events tracked in SMB_ABTSRC

The register increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.

		BASE+0X78					
Bit	31:6	5	4	3	2	1	0
	Reserved	RXFLR					
RST	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:6	Reserved	Reserved.	N/A
5:0	RXFLR	Contains the number of valid data entries in the receive FIFO	R

20.2.2.29 SMB_SDAHD (SMB SDA Hold Time Register)

This register controls the amount of hold time on the SDA signal after a negative edge of SCL in both master and slave mode, in units of APB clock period. The value programmed must be greater than the minimum hold time in each mode for the value to be implemented—one cycle in master mode, seven cycles in slave mode. Writes to this register succeed only when SMB_ENABLE[0]=0.

The programmed SDA hold time cannot exceed at any time the duration of the low part of scl. Therefore the programmed value cannot be larger than N_SCL_LOW-2, where N_SCL_LOW is the duration of the low part of the scl period measured in APB clock cycles.

	SMB_SDAHD	BASE + 0x7C														
Bit	31:16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved	SDAHD															
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:16	Reserved	Reserved.	N/A

15:0	SDAHD	SMB Hold Time. Sets the required SDA hold time in units of APB clock period.	RW

20.2.2.30 SMB_ABTSRC (SMB Transmit Abort Source Register)

This register has 32 bits that indicate the source of the TX_ABRT bit. Except for Bit 9, this register is cleared whenever the SMBCTXABT register or the SMBCINT register is read. To clear Bit 9, the source of the SBYTE_NORSTRT must be fixed first; RESTART must be enabled (SMB_CON[5]=1), the SPECIAL bit must be cleared (SMB_TAR[11]), or the GC_OR_START bit must be cleared (SMB_TAR[10]). Once the source of the SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the SBYTE_NORSTRT is not fixed before attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.

	SMB_ABTSRC																BASE + 0x80													
Bit	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
	TX_FLUSH_CNT								Reserved							USER_ABRT	SLVRD_INTX	SLV_ARBLOST	SLVFLUSH_TXFIFO	ARB_LOST	ABRT_MASTER_DIS	ABRT_10B_RD_NORSTRT	SBYTE_NORSTRT	ABRT_HS_NORSTRT	SBYTE_ACKDET	ABRT_HS_ACKDET	ABRT_GCALL_READ	ABRT_GCALL_NOCK	ABRT_TXDATA_NOACK	ABRT_10ADDR2_NOACK	ABRT_10ADDR1_NOACK	ABRT_7B_ADDR_NOACK
RST	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

Bits	Name	Description	R/W
31:24	TX_FLUSH_CNT	This field preserves the TXFLR value prior to the last TX_ABRT event. It is cleared whenever SMB is disabled.	R
23:17	Reserved	Read as zero.	
16	USER_ABRT	This is a master-mode-only bit. Master has detected the transfer abort	R
15	SLVRD_INTX	1: When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of SMB_DC register.	R
14	SLV_ARBLOST	1: Slave lost the bus while transmitting data to a remote master. SMB_ABTSRC[12] is set at the same time. **NOTE:** Even though the slave never “owns” the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to-high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then SMB no longer own the bus.	R

		Reset value: 0x0.	
13	SLVFLUSH_TXFIFO	1: Slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO. Reset value: 0x0.	R
12	ARB_LOST	1: Master has lost arbitration, or if SMB_ABTSRC[14] is also set, then the slave transmitter has lost arbitration. **NOTE:** SMB can be both master and slave at the same time. Reset value: 0x0.	R
11	ABRT_MASTER_DIS	1: User tries to initiate a Master operation with the Master mode disabled. Reset value: 0x0.	R
10	ABRT_10B_RD_NORSTRT	1: The restart is disabled (RESTART bit (SMB_CON[5]) = 0) and the master sends a read command in 10-bit addressing mode. Reset value: 0x0.	R
9	SBYTE_NORSTRT	To clear Bit 9, the source of the SBYTE_NORSTRT must be fixed first; restart must be enabled (SMB_CON[5]=1), the SPECIAL bit must be cleared (SMB_TAR[11]), or the GC_OR_START bit must be cleared (SMB_TAR[10]). Once the source of the SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets re-asserted. 1: The restart is disabled (RESTART bit (SMB_CON[5]) = 0) and the user is trying to send a START Byte. Reset value: 0x0.	R
8	ABRT_HS_NORSTRT	1: The restart is disabled (RESTART bit (SMB_CON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode. Reset value: 0x0.	R
7	SBYTE_ACKDET	1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior). Reset value: 0x0.	R
6	ABRT_HS_ACKDET	1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior). Reset value: 0x0.	R
5	ABRT_GCALL_READ	1: SMB in master mode sent a General Call but the user programmed the byte following the General Call to be a read from the bus (SMBDC[9] is set to 1). Reset value: 0x0.	R
4	ABRT_GCALL_NOACK	1: SMB in master mode sent a General Call and no slave on the bus acknowledged the General Call. Reset value: 0x0.	R

3	ABRT_TXDATA_NOACK	1: This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledgement from the remote slave(s). Reset value: 0x0.	R
2	ABRT_10ADDR2_NOACK	1: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave. Reset value: 0x0.	R
1	ABRT_10ADDR1_NOACK	1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave. Reset value: 0x0.	R
0	ABRT_7B_ADDR_NOACK	1: Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave. Reset value: 0x0.	R

20.2.2.31 SMB_DMACR (SMB DMA Control Register)

The register is used to enable the DMA Controller interface operation. There is a separate bit for transmit and receive. This can be programmed regardless of the state of SMB_ENABLE.

	SMB_DMACR																BASE + 0x88															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																														TDEN	RDEN
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:2	Reserved	Writing has no effect, read as zero.	R
1	TDEN	**Transmit DMA Enable**. This bit enables/disables the transmit DMA channel. 0: Transmit DMA disabled 1: Transmit DMA enabled Reset value: 0x0.	R/W
0	RDEN	**Receive DMA Enable**. This bit enables/disables the receive DMA channel. 0: Receive DMA disabled 1: Receive DMA enabled Reset value: 0x0.	R/W

20.2.2.32 SMB_DMATDLR (SMB DMA Transmit Data Level Register)

	SMB_DMACR																BASE + 0x8c															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	Reserved										TDLR					
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:5	Reserved	Writing has no effect, read as zero.	R
4:0	TDLR	**DMA Transmit Data Level**. This bit field controls the level at which a DMA request is made by the transmit logic. The dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value.	R/W

20.2.2.33 SMB_DMARDLR (SMB DMA Transmit Data Level Register)

	SMB_DMACR																BASE + 0x90															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	Reserved										RDLR					
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:5	Reserved	Writing has no effect, read as zero.	R
4:0	RDLR	**DMA Receive Data Level**. This bit field controls the level at which a DMA request is made by the receive logic. The dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1.	R/W

20.2.2.34 SMB_SDASU (SMB SDA Setup Register)

This register controls the amount of time delay (in terms of number of APB clock periods) introduced in the rising edge of SCL, relative to SDA changing, when SMB services a read request in a slave-transmitter operation. The relevant SMB requirement is $t_{SU:DAT}$(*NOTE 5*)as detailed in the I2C Bus Specification(Please Google "*UM10204 pdf*").

NOTE: The length of setup time is calculated using [(SMB_SDASU - 1) * (apb_clk_period)], so if the

user requires 10 apb clock periods of setup time, they should program a value of 11. The SMB_SDA_SETUP register is only used by the SMB when operating as a slave transmitter

NOTE 5: A Fast-mode SMB-bus device can be used in a Standard-mode SMB-bus system, but the requirement $t_{SU;DAT}$ >= 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line $t_{r\ max} + t_{SU;DAT}$ = 1000 +250 =1250 ns (according to the Standard-mode SMB-bus specification) before the SCL line is released. Also the acknowledge timing must meet this set-up time(Please refer NXP Semiconductors, *UM10204: I2C-Bus specification and user manual (Rev.6--4 April 2014), Table 10, Fig 38*)

	SMB_SDASU																BASE + 0x94													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								SDASU							
RST																	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0

Bits	Name	Description	R/W
15:8	Reserved	Writing has no effect, read as zero.	R
7:0	SDASU	SDA Setup. It is recommended that if the required delay is 1000ns, then for an APB clock frequency of 10 MHz, SMBSDASU should be programmed to a value of 11.	R/W

20.2.2.35 SMB_ACKGC (SMB ACK General Call Register)

The register controls whether SMB responds with an ACK or NACK when it receives an SMB General Call address.

	SMB_ACKGC																BASE + 0x98													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																															ACKGC
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	Name	Description	R/W
15:1	Reserved	Writing has no effect, read as zero.	R
0	ACKGC	ACK General Call. When set to 1, SMB responds with an ACK when it receives a General Call. When set to 0, the SMB does not generate General Call interrupts.	R/W

20.2.2.36 SMB_ENBST (SMB Enable Status)

The register is used to report the SMB hardware status when the SMB_ENABLE[0] is set from 1 to 0; that is, when SMB is disabled.

If SMB_ENABLE[0] has been set to 1, bits 2:1 are forced to 0, and bit 0 is forced to 1.

If SMB_ENABLE[0] has been set to 0, bits 2:1 is only be valid as soon as bit 0 is read as '0'.

Note: When SMB_ENABLE[0] has been set to 0, a delay occurs for bit 0 to be read as 0 because disabling the SMB depends on SMB bus activities.

	SMB_ENBST															BASE + 0x9C															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																												SLVRDLST	SLVDISB	SMBEN
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
15:3	Reserved	Writing has no effect, read as zero.	R
2	SLVRDLST	Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an SMB transfer due to the setting of SMB_ENABLE[0] from 1 to 0. When read as 1, SMB is deemed to have been actively engaged in an aborted SMB transfer (with matching address) and the data phase of the SMB transfer has been entered, even though a data byte has been responded with a NACK. **NOTE:** If the remote SMB master terminates the transfer with a STOP condition before the SMB has a chance to NACK a transfer, and SMB_ENABLE[0] has been set to 0, then this bit is also set to 1. When read as 0, SMB is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. **NOTE:** The CPU can safely read this bit when SMB_ENABLE[0] is read as 0. Reset value: 0x0.	R
1	SLVDISB	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting of the SMB_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the SMB_ENABLE register while: (a) SMB is receiving the address byte of the Slave-Transmitter operation from a remote master; OR, (b) address and data bytes of the Slave-Receiver operation from a remote master. When read as 1, SMB is deemed to have forced a NACK during any part of an SMB transfer, irrespective of whether the SMB address matches the slave address set in SMB (SMBSAR register) OR if the	R

		transfer is completed before SMBENB is set to 0 but has not taken effect. **NOTES:** 1 If the remote SMB master terminates the transfer with a STOP condition before the SMB has a chance to NACK a transfer, and SMBENB has been set to 0, then this bit will also be set to 1. When read as 0, SMB is deemed to have been disabled when there is master activity, or when the SMB bus is idle. 2 The CPU can safely read this bit when SMB_ENABLE[0] is read as 0. Reset value: 0x0.	
0	SMBEN	When read as 1, SMB is deemed to be in an enabled state. When read as 0, SMB is deemed completely inactive.	R

20.2.2.37 SMB_FLT (SMB Filter Register)

This register is used to store the duration, measured in APB clock cycles, of the longest spike that is filtered out by the spike suppression logic when the component is operating in SS or FS modes. This register must be programmed with a minimum value of 1.

	SMB_FLT																BASE + 0xA0							
Bit																	7	6	5	4	3	2	1	0
																	FLTCNT							
RST																	0	0	0	0	0	0	0	1

Bits	Name	Description	R/W
7:0	FLTCNT	This register sets the duration, measured in SMB device clock cycles, of the longest value in the SCL or SDA lines that are filtered out by the spike suppression logic. This register can be written only when the SMB interface is disabled. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted, results in 1 being set.	RW

20.3 Operating Flow

This section provides information on the following topics:

- "Slave Mode Operation"
- "Master Mode Operation"

– “Disabling SMB”

NOTE: It is important to note that the SMB should only be set to operate as an SMB Master, or SMB Slave, but not both simultaneously. This is achieved by ensuring that bit 6 (SMBSLAVE_DISABLE) and 0 (SMBMASTER_MODE) of the SMBCON register are never set to 0 and 1, respectively.

20.3.1 SMB Behavior

The SMB can be controlled via software to be either:

– An SMB master only, communicating with other SMB slaves.

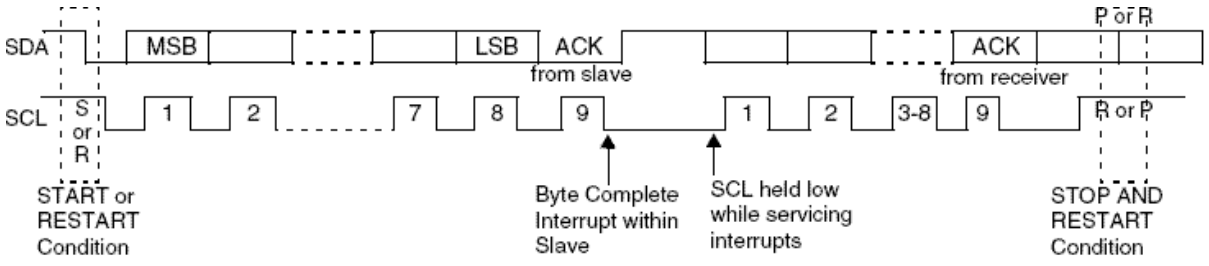
OR

– An SMB slave only, communicating with one more SMB masters.

The master is responsible for generating the clock and controlling the transfer of data. The slave is responsible for either transmitting or receiving data to/from the master. The device that is receiving data, which can be either a master or a slave, sends the acknowledgement of data. As mentioned previously, the SMB protocol also allows multiple masters to reside on the SMB bus and uses an arbitration procedure to determine bus ownership.

Each slave has a unique address that is determined by the system designer. When a master wants to communicate with a slave, the master transmits a START/RESTART condition that is then followed by the slave’s address and a control bit (R/W) to determine if the master wants to transmit data or receive data from the slave. The slave then sends an acknowledgement (ACK) pulse after the address.

If the master (master-transmitter) is writing to the slave (slave-receiver), the receiver gets one byte of data. This transaction continues until the master terminates the transmission with a STOP condition. If the master is reading from a slave (master-receiver), the slave transmits (slave-transmitter) a byte of data to the master, and the master then acknowledges the transaction with the ACK pulse. This transaction continues until the master terminates the transmission by not acknowledging (NACK) the transaction after the last byte is received, and then the master issues a STOP condition or addresses another slave after issuing a RESTART condition.



20.3.2 Master Mode Operation

This section includes the following topics:

- "Initial Configuration"
- "Dynamic SMBTAR or SMB10BITADDR_MASTER Update"
- "Master Transmit and Master Receive"

20.3.2.1 Configuration

To use the SMB as a master perform the following steps:

1 Disable the SMB by writing 0 to the SMBENB register. And wait for the SMBENBST.SMBEN = 0.

2 Write to the SMB_CON register to set the speed mode supported (bits 2:1). Please note that the MATP (bit4) is NOT writable. The addressing mode is controlled by Register SMB_TAR.

3 Set the expected SCL frequency. SMB_CON.SPEED = 2'b01, only SMB_SHCNT and SMBSLCNT are needed to be configured; SMB_CON.SPEED = 2'b10, only SMB_FHCNT and SMB_FLCNT are needed to be configured.

Supposed:

T_{scl} : SMB SCL period

T_{SMB_clk} : SMB device clock period

$T_{min\_scl\_l}$: Protocol minimum SCL low time

$T_{min\_scl\_h}$: Protocol minimum SCL high time

Then can get the equation:

$T_{scl} = T_{SMB\_clk} * ((SMB*HCNT + 8) + (SMB*LCNT + 1))$

And the following conditions should be met:

$(SMB*HCNT + 8) * T_{SMB\_clk} >= T_{min\_scl\_h}$

$(SMB*LCNT + 1) * T_{SMB\_clk} >= T_{min\_scl\_l}$

SMB*HCNT >= 6

SMB*LCNT >= 8

4 Write to the SMB_TAR register the address of the SMB device to be addressed. It also indicates whether a General Call or a START BYTE command is going to be performed by SMB. The addressing mode that master starts, either 7-bit or 10-bit addressing, is controlled by the SMB10BITADDR_MASTER bit field (bit 12).

5 Enable the SMB by writing a 1 into SMB_ENABLE register. And wait for the SMB_ENBST.SMBEN = 1.

6 Now write the transfer direction and data to be sent to the SMB_DC register (This can be done by DMA). If the SMB_DC register is written before the SMB is enabled, the data and commands are lost as the buffers are kept cleared when SMB is not enabled.

NOTE: For multiple SMB transfers, perform additional writes to the TX FIFO such that the TX FIFO does not become empty during the SMB transaction. If the TX FIFO is completely emptied at any stage and SMB_CON.STPHLD is 0, then further writes to the TX FIFO results in an independent SMB transaction.

20.3.2.2 Dynamic SMBTAR or SMB10BITADDR_MASTER Update

The SMB supports dynamic updating of the SMBTAR (bits 9:0) and SMB10BITADDR_MASTER (bit 12) bit fields of the SMBTAR register. You can dynamically write to the SMBTAR register provided the following conditions are met:

1 SMB is not enabled (SMBENB=0);

OR

2 SMB is enabled (SMBENB=1);
AND
SMB is NOT engaged in any Master (tx, rx) operation (SMBST[5]=0);
AND
SMB is enabled to operate in Master mode (SMBCON[0]=1);
AND
there are NO entries in the TX FIFO (SMBST[2]=0).

20.3.2.3 Master Transmit and Master Receive

The SMB supports switching back and forth between reading and writing dynamically. To transmit data, write the data to be transferred to lower byte of the SMB Rx/Tx Data Buffer and Command Register (SMB_DC). The *CMD* bit (SMB_DC[8]) should be written to 0 for SMB write operations.

Subsequently, a read command may be issued by writing “don’t cares” to the lower byte of the SMBDC register, and a 1 should be written to the *CMD* bit.

20.3.3 Slave Mode Operation

This section includes the following procedures:

- “Initial Configuration”
- “Slave-Transmitter Operation for a Single Byte”
- “Slave-Receiver Operation for a Single Byte”
- “Slave-Transfer Operation For Bulk Transfers”

20.3.3.1 Initial Configuration

To use the SMB as a slave, perform the following steps:

1 Disable the SMB by writing a ‘0’ to bit 0 of the SMBENB register.
2 Write to the SMB_SAR register (bits 9:0) to set the slave address. This is the address to which the SMB responds.
3 Write to the SMB_CON register to specify which type of addressing is supported (7- or 10-bit by setting bit 3). Enable the SMB in slave-only mode by writing a ‘0’ into bit 6 (SMBSLAVE_DISABLE) and a ‘0’ to bit 0 (MASTER_MODE).

NOTE: Slaves and masters do not have to be programmed with the same type of addressing 7- or 10-bit address. For instance, a slave can be programmed with 7-bit addressing

and a master with 10-bit addressing, and vice versa.

4 Enable the SMB by writing a '1' in bit 0 of the SMB_ENB register.

NOTE: Depending on the reset values chosen, steps 2 and 3 may not be necessary because the reset values can be configured. For instance, if the device is only going to be a master, there would be no need to set the slave address because you can configure SMB to have the slave disabled after reset and to enable the master after reset. The values stored are static and do not need to be reprogrammed if the SMB is disabled.

20.3.3.2 Slave-Transmitter Operation for a Single Byte

When another SMB master device on the bus addresses the SMB and requests data, the SMB acts as a slave-transmitter and the following steps occur:

1 The other SMB master device initiates an SMB transfer with an address that matches the slave address in the SMB_SAR register of the SMB.

2 The SMB acknowledges the sent address and recognizes the direction of the transfer to indicate that it is acting as a slave-transmitter.

3 The SMB asserts the RDREQ interrupt (bit 5 of the SMBINTST register) and holds the SCL line low. It is in a wait state until software responds.
If the RDREQ interrupt has been masked, due to SMB_INTM[5] register (MRDREQ bit field) being set to 0, then it is recommended that a hardware and/or software timing routine be used to instruct the CPU to perform periodic reads of the SMB_INTST register.
   a Reads that indicate SMB_INTST[5] (RDREQ bit field) being set to 1 must be treated as the equivalent of the RDREQ interrupt being asserted.
   b Software must then act to satisfy the SMB transfer.
   c The timing interval used should be in the order of 10 times the fastest SCL clock period the SMB can handle. For example, for 400 kb/s, the timing interval is 25us.

NOTE: The value of 10 is recommended here because this is approximately the amount of time required for a single byte of data transferred on the SMB bus.

4 If there is any data remaining in the TX FIFO before receiving the read request, then the SMB asserts a TX_ABRT interrupt (bit 6 of the SMBINTST register) to flush the old data from the TX FIFO.

NOTE: Because the SMB's TX FIFO is forced into a flushed/reset state whenever a TX_ABRT event occurs, it is necessary for software to release the SMB from this state by reading the SMB_CTXABT register before attempting to write into the TX FIFO. See register SMB_INTST for more details. If the TX_ABRT interrupt has been masked, due to of SMB_INTM[6] register (MTX_ABRT bit field) being set to 0, then it is recommended that re-using the timing routine (described in the previous step), or a similar one, be used to read the SMBINTST register.

a Reads that indicate bit 6 (TXABT) being set to 1 must be treated as the equivalent of the TX_ABRT interrupt being asserted.
b There is no further action required from software.
c The timing interval used should be similar to that described in the previous step for the SMBINTST[5] register.

5 Software writes to the SMB_DC register with the data to be written (by writing a '0' in bit 8).
6 Software must clear the RDREQ and TX_ABRT interrupts (bits 5 and 6, respectively) of the SMBINTST register before proceeding.
If the RDREQ and/or TX_ABRT interrupts have been masked, then clearing of the SMB_INTST register will have already been performed when either the RDREQ or TXABT bit has been read as 1.
7 The SMB releases the SCL and transmits the byte.
8 The master may hold the SMB bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

20.3.3.3 Slave-Receiver Operation for a Single Byte

When another SMB master device on the bus addresses the SMB and is sending data, the SMB acts as a slave-receiver and the following steps occur:

1 The other SMB master device initiates an SMB transfer with an address that matches the SMB's slave address in the SMBSAR register.
2 The SMB acknowledges the sent address and recognizes the direction of the transfer to indicate that the SMB is acting as a slave-receiver.
3 SMB receives the transmitted byte and places it in the receive buffer.

NOTE: If the RX FIFO is completely filled with data when a byte is pushed, then an overflow occurs and the SMB continues with subsequent SMB transfers. Because a NACK is not generated, software must recognize the overflow when indicated by the SMB (by the RXOF bit in the SMBINTST register) and take appropriate actions to recover from lost data. Hence, there is a real time constraint on software to service the RX FIFO before the latter overflow as there is no way to re-apply pressure to the remote transmitting master. You must select a deep enough RX FIFO depth to satisfy the interrupt service interval of their system.

4 SMB asserts the RX_FULL interrupt (SMBINTST[2] register).
If the RX_FULL interrupt has been masked, due to setting SMB_INTM[2] register to 0 or setting SMB_TX_TL to a value larger than 0, then it is recommended that a timing routine be implemented for periodic reads of the "SMBST". Reads of the SMB_ST register, with bit 3 (RFNE) set at 1, must then be treated by software as the equivalent of the RX_FULL interrupt being asserted.
5 Software may read the byte from the SMB_DC register (bits 7:0).

6 The other master device may hold the SMB bus by issuing a RESTART condition or release the bus by issuing a STOP condition.

20.3.3.4 Slave-Transfer Operation For Bulk Transfers

In the standard SMB protocol, all transactions are single byte transactions and the programmer responds to a remote master read request by writing one byte into the slave's TX FIFO.

When a slave (slave-transmitter) is issued with a read request (RDREQ) from the remote master (master-receiver), at a minimum there should be at least one entry placed into the slave-transmitter's TX FIFO.

SMB is designed to handle more data in the TX FIFO so that subsequent read requests can take that data without raising an interrupt to get more data. Ultimately, this eliminates the possibility of significant latencies being incurred between raising the interrupt for data each time had there been a restriction of having only one entry placed in the TX FIFO.

This mode only occurs when SMB is acting as a slave-transmitter. If the remote master acknowledges the data sent by the slave-transmitter and there is no data in the slave's TX FIFO, the SMB holds the SMB SCL line low while it raises the read request interrupt (RDREQ) and waits for data to be written into the TX FIFO before it can be sent to the remote master.

If the RDREQ interrupt is masked, due to bit 5 (MRDREQ) of the SMB_INTST register being set to 0, then it is recommended that a timing routine be used to activate periodic reads of the SMB_INTST register. Reads of SMB_INTST that return bit 5 (RDREQ) set to 1 must be treated as the equivalent of the RDREQ interrupt referred to in this section.

The RDREQ interrupt is raised upon a read request, and like interrupts, must be cleared when exiting the interrupt service handling routine (ISR). The ISR allows you to either write 1 byte or more than 1 byte into the TX FIFO. During the transmission of these bytes to the master, if the master acknowledges the last byte. Then the slave must raise the RDREQ again because the master is requesting for more data.

20.3.4 Disabling SMB

The register SMB_ENBST is added to allow software to unambiguously determine when the hardware has completely shutdown in response to the SMB_ENABLE register being set from 1 to 0.

20.3.4.1 Procedure

1 Define a timer interval (tSMB_poll) equal to the 10 times the signaling period for the highest SMB transfer speed used in the system and supported by SMB. For example, if the highest SMB transfer mode is 400 kb/s, then this tSMB_poll is 25us.

2 Define a maximum time-out parameter, MAX_T_POLL_COUNT, such that if any repeated polling operation exceeds this maximum value, an error is reported.

3 Execute a blocking thread/process/function that prevents any further SMB master transactions to be started by software, but allows any pending transfers to be completed.
NOTE: This step can be ignored if SMB is programmed to operate as an SMB slave only.

4 The variable POLL_COUNT is initialized to zero.

5 Set SMBENB to 0.

6 Read the SMB_ENBST register and test the SMB_ENABLE[0]. Increment POLL_COUNT by one. If POLL_COUNT >= MAX_T_POLL_COUNT, exit with the relevant error code.

7 If SMB_ENBST[0] is 1, then sleep for tSMB_poll and proceed to the previous step. Otherwise, exit with a relevant success code.

20.3.5 Summary the condition could flush TX FIFO

1. When IC_ENABLE[0] is set to 0, the TX FIFO is flushed and held in reset.
2. TX_ABRT event occurs. The TX FIFO remains in this flushed state until the register SMB_CTXABT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface.
3. When act as a slave, the slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO.

20.3.6 The condition could generate START, STOP and RESTART

A STOP condition is generated only when the user specifically requests it by setting bit 9 (Stop bit) of the command written to SMB_DC register. Naturally, if the last byte send out does not set bit 9 (Stop bit), the component does not generate a STOP if the Tx FIFO becomes empty; in this situation the component holds the SCL line low, stalling the bus until a new entry is available in the Tx FIFO

Figure 20-1 illustrates the behavior of the SMB when the Tx FIFO becomes empty while operating as a master transmitter, as well as showing the generation of a STOP condition.

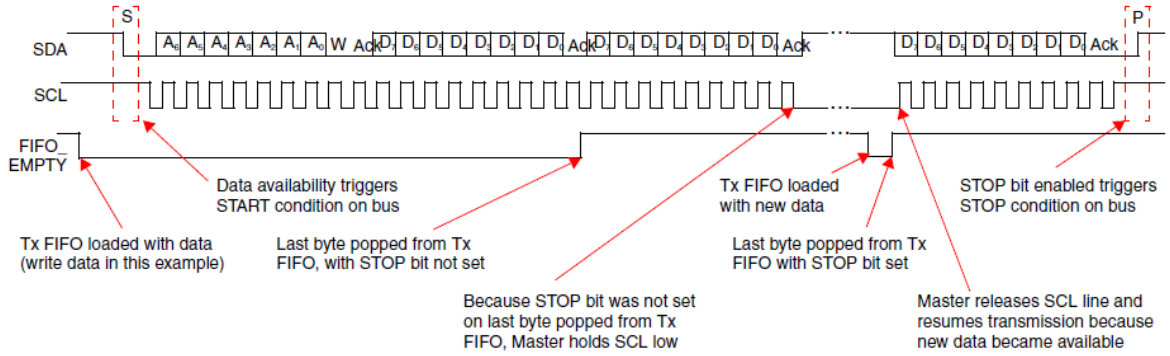


Figure 20-1 Master Transmitter — Tx FIFO Empties/STOP Generation

Figure 20-2 illustrates the behavior of the SMB when the Tx FIFO becomes empty while operating as a master receiver, as well as showing the generation of a STOP condition.

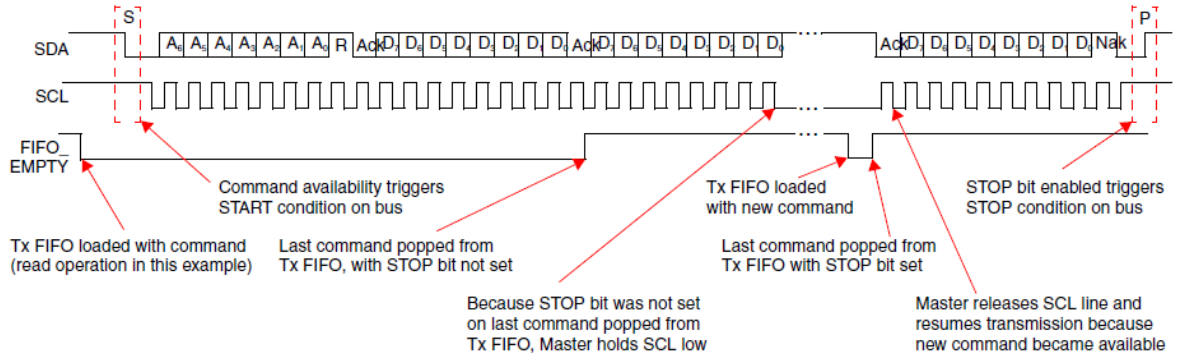


Figure 20-2 Master Receiver — Tx FIFO Empties/STOP Generation

Figure 20-3 and Figure 20-4 illustrate configurations where the user can control the generation of RESTART conditions on the SMB bus. If bit 10 (Restart) of the SMB_DC register is set and the restart capability is enabled (IC_RESTART_EN=1), a RESTART is generated before the data byte is written to or read from the slave. If the restart capability is not enabled a STOP followed by a START is generated in place of the RESTART. Figure 20-3 illustrates this situation during operation as a master transmitter.

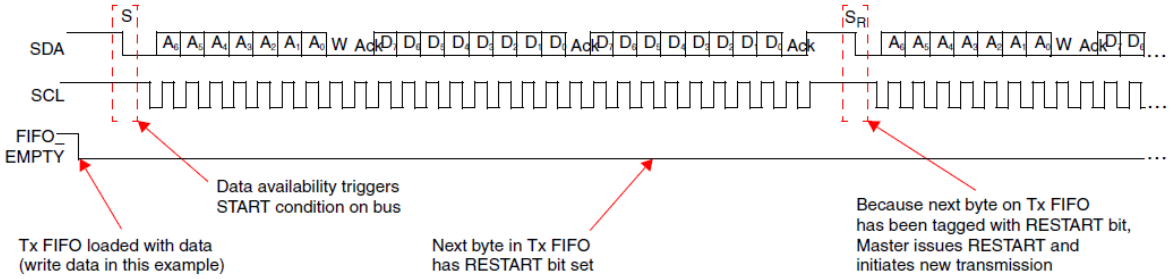


Figure 20-3 Master Transmitter — Restart bit of SMB_DC is set

Figure 20-4 illustrates the same situation, but during operation as a master receiver.

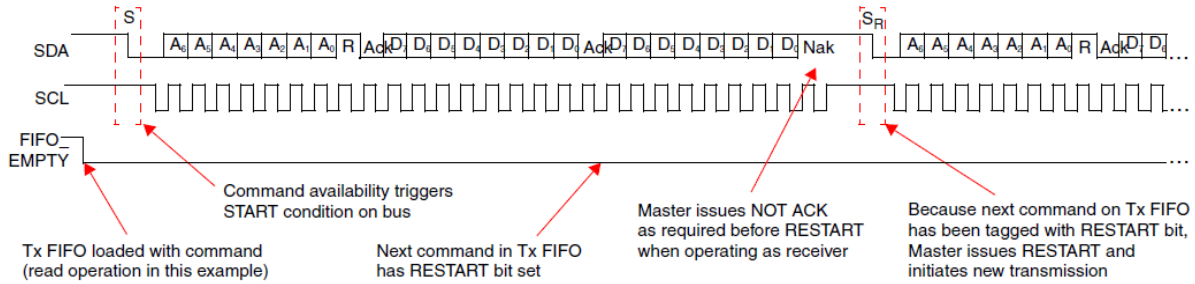


Figure 20-4 Master Receiver — Restart bit of SMB_DC is set

Figure 20-5 illustrates operation as a master transmitter where the Stop bit of the SMB_DC register is set and the Tx FIFO is not empty

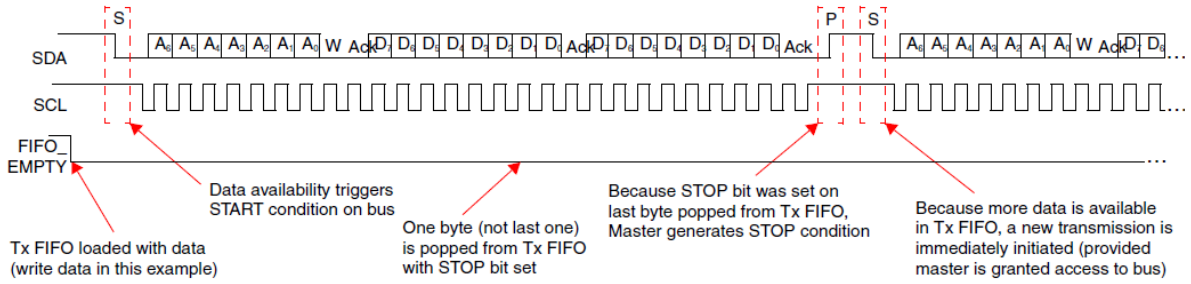


Figure 20-5 Master Transmitter — Stop Bit of SMB_DC Set/Tx FIFO Not Empty

Figure 20-6 illustrates operation as a master transmitter where the first byte loaded into the Tx FIFO is allowed to go empty with the Restart bit set.

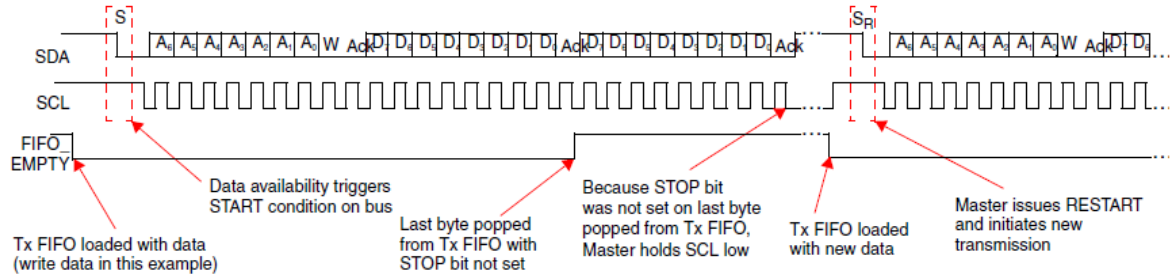


Figure 20-6 Master Transmitter — First Byte Loaded Into Tx FIFO Allowed to Empty, Restart Bit Set

Figure 20-7 illustrates operation as a master receiver where the Stop bit of the SMB_DC register is set and the Tx FIFO is not empty

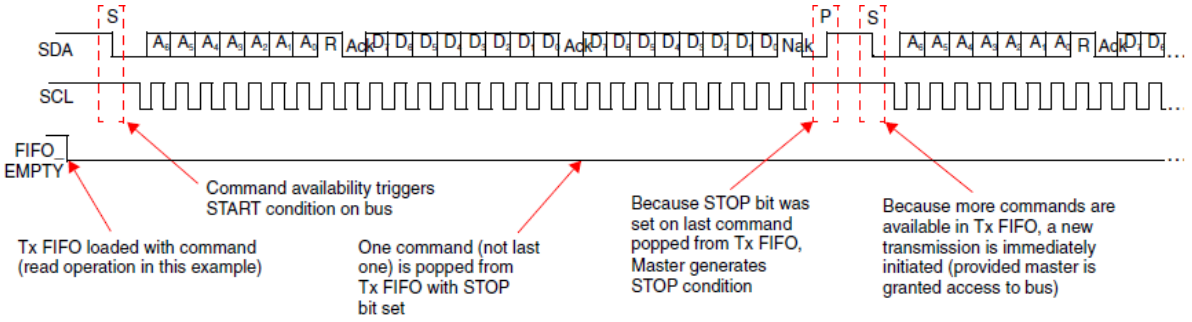


Figure 20-7 Master Receiver — Stop Bit of IC_DATA_CMD Set/Tx FIFO Not Empty

Figure 20-8 illustrates operation as a master receiver where the first command loaded after the Tx FIFO is allowed to empty and the Restart bit is set

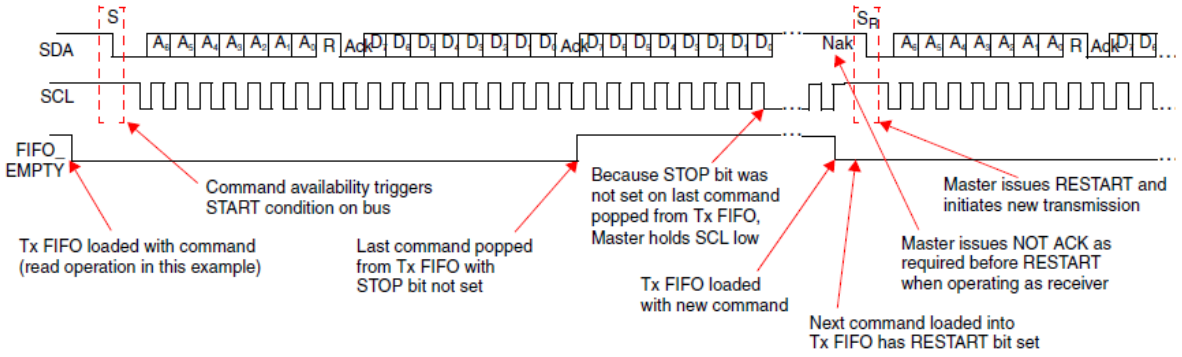


Figure 20-8 Master Receiver — First Command Loaded After Tx FIFO Allowed to Empty/Restart Bit Set

21 Smart Card Controller

21.1 Overview

Smart Card Controller (SCC) interface is a primary device and communications interface for kinds of IC cards. The SCC interface supports communication with smart cards as specified in standard ISO7816-3. There are two SCC interfaces integrated in this SOC. And they perform the same functions. The SCC interface supports T=0 and T=1 protocol defined in ISO7816-3.

Software controls the session between the SCC interface and the card by SCC registers. Choosing protocol type and parameters, receiving and sending a byte to/from the card, activating/deactivating the card, and similar operations are accomplished with read/write operations to SCC registers. Transforming byte convention (inverse to direct and vice-versa, according to the session convention) is performed within SCC. Hence, software does not have to perform format inversion before character receipt. The SCC interface provides functionality to support the above standards, but it is the responsibility of software to ensure the standards are met.

Features:

- Supports normal card and UIM card.
- 8-bit, 16-level receive-/transmit- FIFO.
- Supports asynchronous character (T=0) communication modes.
- Supports asynchronous block (T=1) communication modes.
- Supports setting of clock-rate conversion factor F (372, 512, 558, etc.), and bit-rate adjustment factor D (1, 2, 4, 8, 16, 32, 12, 20, etc.).
- Supports extra guard time waiting.
- Auto-error detection in T=0 receive mode.
- Auto-character repeat in T=0 transmit mode.
- Transforms inverted format to regular format and vice versa.
- Support stop clock function in some power consuming sensitive applications.

21.2 Pin Description

Table 21-1 Smart Card Controller Pins Description

Name	I/O	Description
SCC_CLK	Output	Serial clock connects SCC and the card.
SCC_DAT	Input/Output	Data communication pin.

21.3 Register Description

Conventions:

5. Register Address = Base + Address offset
6. The registers can be read and written by APB bus
7. Register read/write attribute

   R - Read only

   W - Write only

   RW - read and write

   RCW - read and write, but clear to 0 by read

   RSW - read and write, but set to 1 by read

   RWC - read and write, clear to 0 by write 1, write 0 has no effect

   RWS - read and write, set to 1 by write 1, write 0 has no effect

   RC - read only, and clear to 0 by read

   RS - read only, and set to 1 by read

   SPEC - special access method, relate to its description
8. Reset Value

   1 - reset to 1

   0 - reset to 0

   ? - value unknown after reset

Table 21-2 Smart Card Controller Registers Description

Name	RW	Reset Value	Address	Access Size
SCCDR	RW	0x??	0x10040000	8
SCCFDR	R	0x00	0x10040004	8
SCCCR	RW	0x00000000	0x10040008	32
SCCSR	RW	0x8000	0x1004000C	16
SCCTFR	RW	0x0173	0x10040010	16
SCCEGTR	RW	0x00	0x10040014	8
SCCECR	RW	0x00000000	0x10040018	32
SCCRTOR	RW	0x00	0x1004001C	8

21.3.1 Transmit/Receive FIFO Data Register (SCCDR)

	SCCDR																												0x10040000	
Bit																									7	6	5	4	3	2	1	0
																									DR							
RST																									?	?	?	?	?	?	?	?

Bits	Name	Description	RW
7:0	DR	Data port of HW FIFO.	RW

21.3.2 FIFO Data Count Register (SCCFDR)

	SCCFDR																												0x10040004	
Bit																									7	6	5	4	3	2	1	0
																									Reserved			R				
RST																									0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:5	Reserved		R
4:0	R	Characters resisted in FIFO	R

21.3.3 Control Register (SCCCR)

	SCCCR																												0x10040008	
Bit	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
	SCCE	TRS	T2R	Reserved			FDIV		FLUSH	Reserved					TRIG		TP	CONV	TXIE	RXIE	TENDIE	RTOIE	ECIE	EPIE	RETIE	EOIE	Reserved		TSEND	PX		CLKSTP
RST	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

Bits	Name	Description	RW
31	SCCE	Enables or disables SCC. When disable it, the SCC_CLK will be stopped.	RW
30	TRS	Transmit or Receive Select. 0 – Reception mode. 1 – Transmission mode.	RW
29	T2R	Auto-T2R support. T2R means Transmit turn to Reception. 0 – controlled by SW. 1 – controlled by HW.	RW
28:26	Reserved		R

<table>
<tr><td>25:24</td><td>FDIV</td><td>Frequency Divider Select.<table><tr><th></th><th>SCC_CLK frequency</th></tr><tr><td>00</td><td>Same as device clk</td></tr><tr><td>01</td><td>Half of device clk</td></tr><tr><td>10</td><td>¼ of device clk</td></tr><tr><td>11</td><td>Reserved</td></tr></table></td><td>RW</td></tr>
<tr><td>23</td><td>FLUSH</td><td>Flush FIFO. 0 – Does not empty the Rx/Tx FIFO. 1 – Empty the Rx/Tx FIFO.</td><td>RW</td></tr>
<tr><td>22:18</td><td>Reserved</td><td></td><td>R</td></tr>
<tr><td>17:16</td><td>TRIG</td><td>Receive/Transmit FIFO trigger.<table><tr><th></th><th>Trigger Value</th></tr><tr><td>00</td><td>1</td></tr><tr><td>01</td><td>4</td></tr><tr><td>10</td><td>8</td></tr><tr><td>11</td><td>14</td></tr></table></td><td>RW</td></tr>
<tr><td>15</td><td>TP</td><td>Communicate protocol. 0 – T=0, 1 – T=1</td><td>RW</td></tr>
<tr><td>14</td><td>CONV</td><td>Card data transfer convention. 0 – LSB first. 1 – MSB first and inverted.</td><td>RW</td></tr>
<tr><td>13</td><td>TXIE</td><td>Tx FIFO counter meets the trigger value interrupt enable bit.</td><td>RW</td></tr>
<tr><td>12</td><td>RXIE</td><td>Rx FIFO counter meets the trigger value interrupt enable bit.</td><td>RW</td></tr>
<tr><td>11</td><td>TENDIE</td><td>Transmission finished interrupt enable bit. (Both FIFO and transmitter are empty)</td><td>RW</td></tr>
<tr><td>10</td><td>RTOIE</td><td>Reception timeout interrupt enable bit.</td><td>RW</td></tr>
<tr><td>9</td><td>ECIE</td><td>ETU counter overflow interrupt enable bit.</td><td>RW</td></tr>
<tr><td>8</td><td>EPIE</td><td>Parity error interrupt enable bit.</td><td>RW</td></tr>
<tr><td>7</td><td>RETIE</td><td>Re-transmitting 3 times interrupt enable bit.</td><td>RW</td></tr>
<tr><td>6</td><td>EOIE</td><td>Receive overrun error interrupt enable bit.</td><td>RW</td></tr>
<tr><td>5:4</td><td>Reserved</td><td></td><td>R</td></tr>
<tr><td>3</td><td>TSEND</td><td>Receive TS character stage finished bit. 0 – TS character has not been received. 1 – TS character has been received.</td><td>RW</td></tr>
<tr><td>2:1</td><td>PX</td><td>Parameter X. SCC stop clock mode selection.<table><tr><th></th><th>SCC clock stop</th></tr><tr><td>00</td><td>Does not support SCC clock stop.</td></tr><tr><td>01</td><td>SCC_CLK stops at state low.</td></tr><tr><td>10</td><td>SCC_CLK stops at state high.</td></tr><tr><td>11</td><td>Reserved</td></tr></table></td><td>RW</td></tr>
<tr><td>0</td><td>CLKSTP</td><td>SCC clock stop. 0 – SCC has left or is leaving clock stop mode. 1 – SCC has entered or is entering clock stop mode.</td><td>RW</td></tr>
</table>

21.3.4 Status Register (SCCSR)

	SCCSR															0x1004000C															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																TRANS	Reserved		ORER	RTO	PER	TFTG	RFTG	TEND	Reserved		RETR_3	Reserved			ECNTO
RST																1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15	TRANS	Transfer status. Specifies whether the transfer is stopped or not.	R
14:13	Reserved		R
12	ORER	Receive overrun error.	RW
11	RTO	Reception timeout.	R
10	PER	Parity error.	RW
9	TFTG	Hit Tx FIFO trigger.	R
8	RFTG	Hit Rx FIFO trigger.	R
7	TEND	Transmission end. Both Tx FIFO and transmitter are empty.	RW
6:5	Reserved		R
4	RETR_3	Re-transmit exceed 3 times.	RW
3:1	Reserved		R
0	ECNTO	ETU counter overflow.	RW

21.3.5 Transmission Factor Register (SCCTFR)

	SCCTFR															0x10040010															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																Reserved					FD										
RST																0	0	0	0	0	0	0	1	0	1	1	1	0	0	1	1

Bits	Name	Description	RW
15:11	Reserved		R
10:0	FD	Value of F/D. The initial value is 0x173(371).	RW

21.3.6 Extra Guard Timer Register (SCCEGTR)

	SCCEGTR																												0x10040014	
Bit																									7	6	5	4	3	2	1	0
																									EGTR							
RST																									0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:0	EGTR	The register is corresponding with N value of ISO7816-3. Which indicates the extra-guard time of a transmission.	RW

21.3.7 ETU Counter Value Register (SCCECR)

	SCCECR																												0x10040018	
Bit	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
	Reserved												EC																			
RST	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

Bits	Name	Description	RW
31:20	Reserved		R
19:0	EC	ETU counter. Write operation will clear the internal counter automatically.	RW

21.3.8 Reception Timeout Register (SCCRTOR)

	SCCRTOS																												0x1004001C	
Bit																									7	6	5	4	3	2	1	0
																									RTOR							
RST																									0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:0	RTO	Retry times when parity error detected.	RW

22 Synchronous Serial Interface(SSI)

22.1 Overview

The SSI is a full-duplex synchronous serial interface and can connect to a variety of external analog-to-digital (A/D) converters, audio and telecom CODECS, and other devices that use serial protocols for transferring data. The SSI supports National's Microwire, Texas Instruments Synchronous Serial Protocol (SSP), and Motorola's Serial Peripheral Interface (SPI) protocol.

The SSI operates in master mode (the attached peripheral functions as a slave) and supports serial bit rates from 7.2 KHz to 50 MHz. Serial data formats may range from 2 to 32 bits in length. The SSI provides 128 entries deep x 32 bits wide transmit and receive data FIFOs.

The FIFOs may be loaded or emptied by the Central Processor Unit (CPU) using programmed I/O, or DMA transfers while receiving or transmitting.

22.2 Features:

- 3 protocols support: National's Microwire, TI's SSP, and Motorola's SPI
- Full-duplex or transmit-only or receive-only operation
- Programmable transfer order: MSB first or LSB first
- 128 entries deep x 32 bits wide transmit and receive data FIFOs
- Configurable normal transfer mode or Interval transfer mode
- Programmable clock phase and polarity for Motorola's SSI format
- Two slave select signal (SSI_CE0 / SSI_CE1) supporting up to 2 slave devices
- Back-to-back character transmission/reception mode
- Loop back mode for testing

22.3 Block Diagram

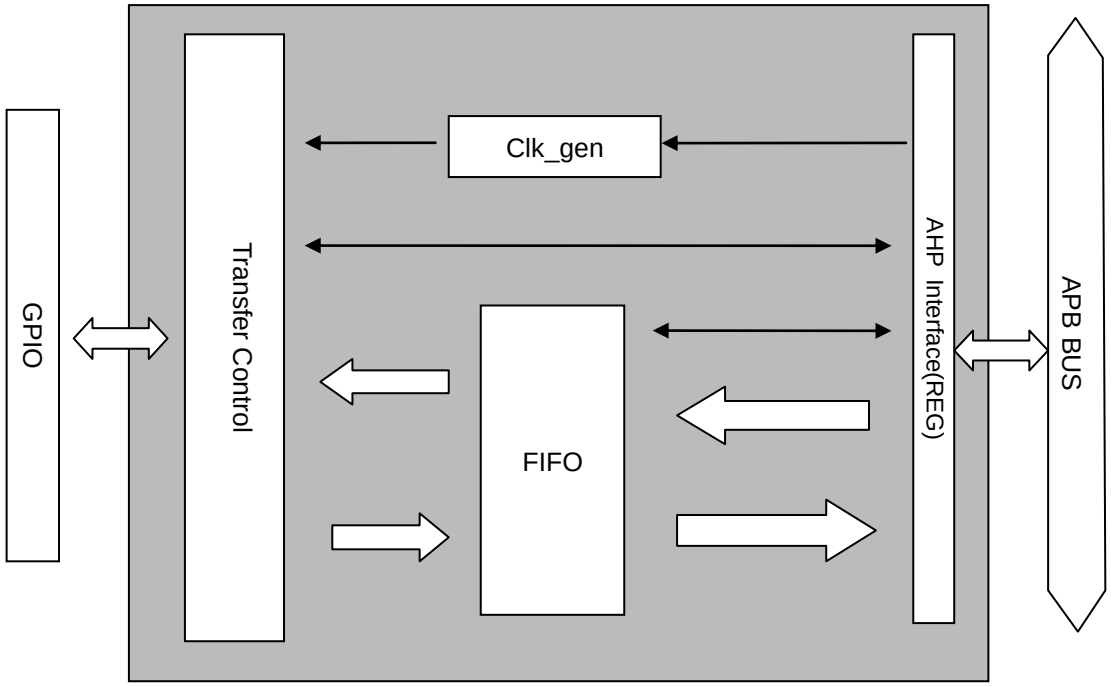


Figure 22-1 SSI Block Diagram

The APB Interface is a slave interface. It is used for register configuration and data transfer.
The FIFO block is bidirectional. There are two 128x32 memory in this block, one used as transmission data buffer, and the other used for data reception.
The Clk_gen module generate the sub-clocks for the Transfer Control module.
The Transfer Control act as the handler of SPI timing generation.

22.4 Functional Description

Serial data is transferred between the processor and external peripheral through FIFO buffers in the SSI. Data transfers to system memory are handled by either the CPU (using programmed I/O) or by DMA. Operation is full duplex - separate buffers and serial data paths permit simultaneous transfers to and from the external peripheral.

Programmed I/O transmits and receives data directly between the CPU and the transmit/receive FIFO's. The DMA controller transfers data during transmit and receive operations between memory and the FIFO's.

Transmit data is written by the CPU or DMA to the SSI's transmit FIFO. The SSI then takes the data from the FIFO, serializes it, and transmits it via the SSI_DT signal to the peripheral. Data from the peripheral is received via the SSI_DR signal, converted to parallel words and is stored in the Receive FIFO. Read operations automatically target the receive FIFO, while write operations write data to the transmit FIFO. Both the transmit and receive FIFO buffers are 128 entries deep by 17 bits wide. As the

received data fills the receive FIFO, a programmable threshold triggers an interrupt to the Interrupt Controller. If enabled, an interrupt service routine responds by identifying the source of the interrupt and then performs one or several read operations from the inbound (receive) FIFO buffer.

22.5 Pin Description

Table 22-1 SSI Controller Pins Description

Name	I/O	Description
SSI_CLK	Output	Serial bit-rate clock
SSI_CE0	Output	First slave select enable
SSI_CE1	Output	Second slave select enable
SSI_GPC	Output	General purpose control signal to external chip
SSI_DT	Output	Transmit data (serial data out)
SSI_DR	Input	Receive data (serial data in)

SSI_CLK is the bit-rate clock driven from the SSI to the peripheral. SSI_CLK is toggled only when data is actively being transmitted and received.
SSI_CE0 or SSI_CE1 are the frame signal, indicating the begin and the end of a serialized data word.
SSI_DT and SSI_DR are the Transmit and Receive serial data lines.
SSI_GPC is general-purpose control signal, synchronized with SSI_CLK, can be used for LCD control.

22.6 Data Formats

Four signals are used to transfer data between the processor and external peripheral. The SSI supports three formats: Motorola SPI, Texas Instruments SSP, and National Microwire. Although they have the same basic structure the three formats have significant differences, as described below:

1. SSI_CE0/SSI_CE1 varies for each protocol as follows:
   - For SPI and Microwire formats, SSI_CE0/SSI_CE1 functions as a chip select to enable the external device (target of the transfer), and is held active-low during the data transfer.
   - For SSP format, this signal is pulsed high for one serial bit-clock period at the start of each frame.
2. SSI_CLK varies for each protocol as follows:
   - For Microwire, both transmit and receive data sources switch data on the falling edge of SSI_CLK, and sample incoming data on the rising edge.
   - For SSP, transmit and receive data sources switch data on the rising edge of SSI_CLK, and sample incoming data on the falling edge.
   - For SPI, the user has the choice of which edge of SSI_CLK to use for switching outgoing data, and for sampling incoming data. In addition, the user can move the phase of SSI_CLK, shifting its active state one-half period earlier or later at the start and end of a frame.

While SSP and SPI are full-duplex protocols, Microwire uses a half-duplex master-slave messaging protocol. At the start of a frame, a 1 or 2-byte control message is transmitted from the controller to the

peripheral. The peripheral does not send any data. The peripheral interprets the message and, if it is a READ request, responds with requested data, one clock after the last bit of the requesting message. The serial clock (SSI_CLK) only toggles during an active frame. At other times it is held in an inactive or idle state, as defined by its specified protocol.

22.6.1 Motorola's SPI Format Details

22.6.1.1 General Single Transfer Formats

The figures below show the timing of general single transfer format.

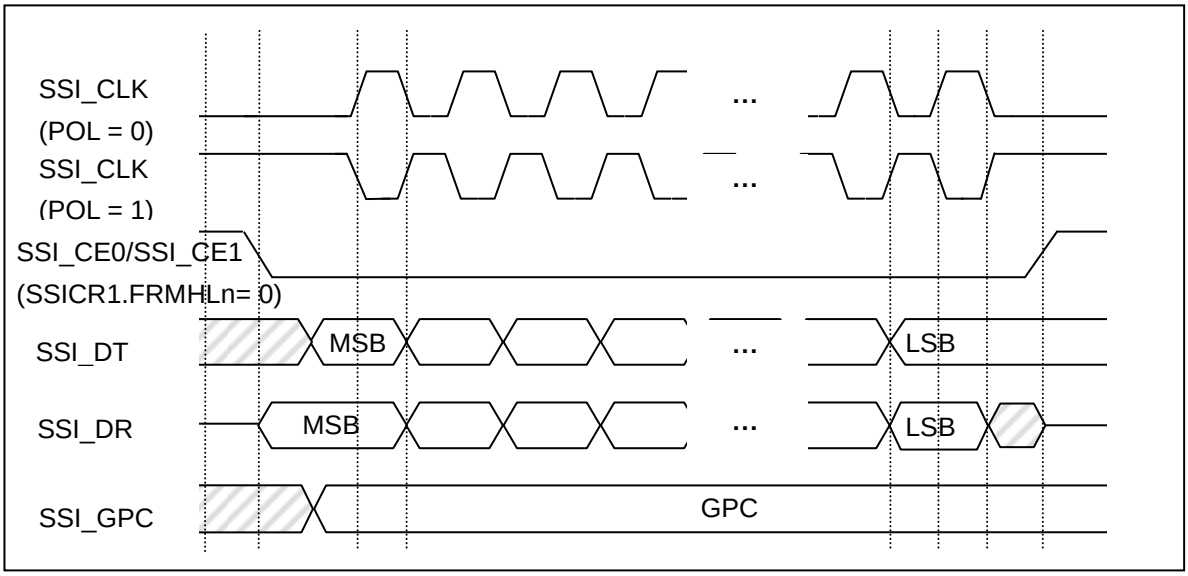


Figure 22-2 SPI Single Character Transfer Format (PHA = 0)

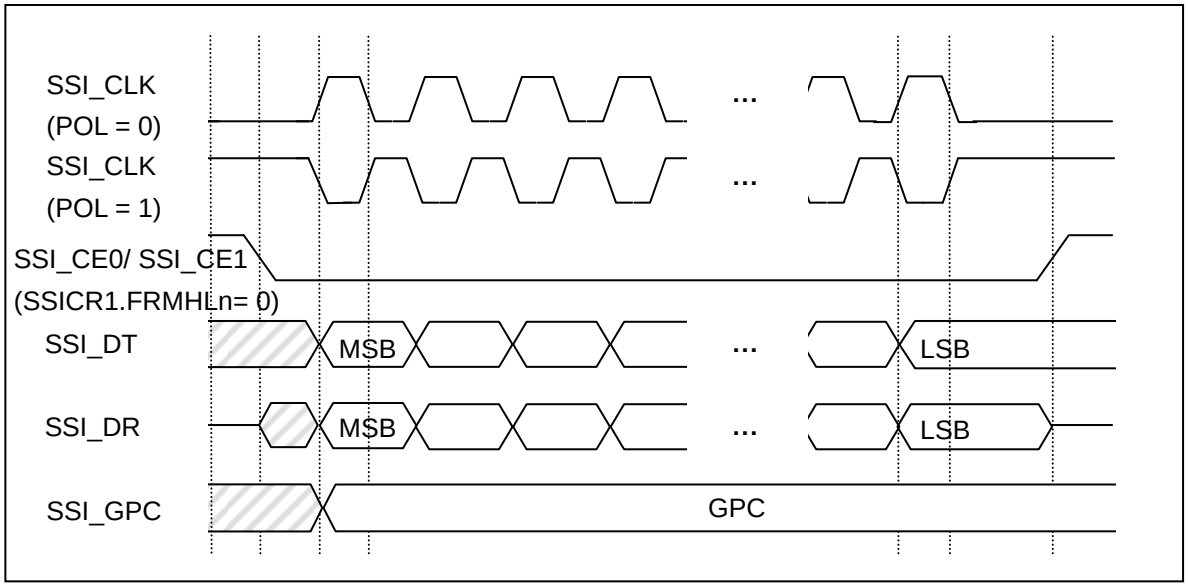


Figure 22-3 SPI Single Character Transfer Format (PHA = 1)

For SSICR1.PHA = 0, when SSICR1.TFVCK = B'00, hardware ensures the first clock edge appears one SSI_CLK period after SSI_CE0 / SSI_CE1 goes valid; when SSICR1.TCKFI = B'00, hardware ensures the SSI_CE0 / SSI_CE1 negated half SSI_CLK period after last clock change edge; when SSICR1.TFVCK ≠ B'00 or SSICR1.TCKFI ≠ B'00, 1/2/3 more clock cycles are inserted.

For SSICR1.PHA = 1, when SSICR1.TFVCK = B'00, hardware ensures the first clock edge appears half SSI_CLK period after SSI_CE0 / SSI_CE1 goes valid; when SSICR1.TCKFI = B'00, hardware ensures the SSI_CE0 / SSI_CE1 negated one SSI_CLK period after last clock change edge; when SSICR1.TFVCK ≠ B'00 or SSICR1.TCKFI ≠ B'00, 1/2/3 more clock cycles are inserted.
Data is sampled from SSI_DR at every rising edge (when PHA = 0, POL = 0 or PHA = 1, POL = 1) or at every falling edge (when PHA = 0, POL = 1 or PHA = 1, POL = 0). According to SPI protocol, input data on SSI_DR should be stable at every sample clock edge.

Drive data onto SSI_DT at every rising edge (when PHA = 0, POL = 1 or PHA = 1, POL = 0) or at every falling edge (when PHA = 0, POL = 0 or PHA = 1, POL = 1).

22.6.1.2 Back-to-Back Transfer Formats

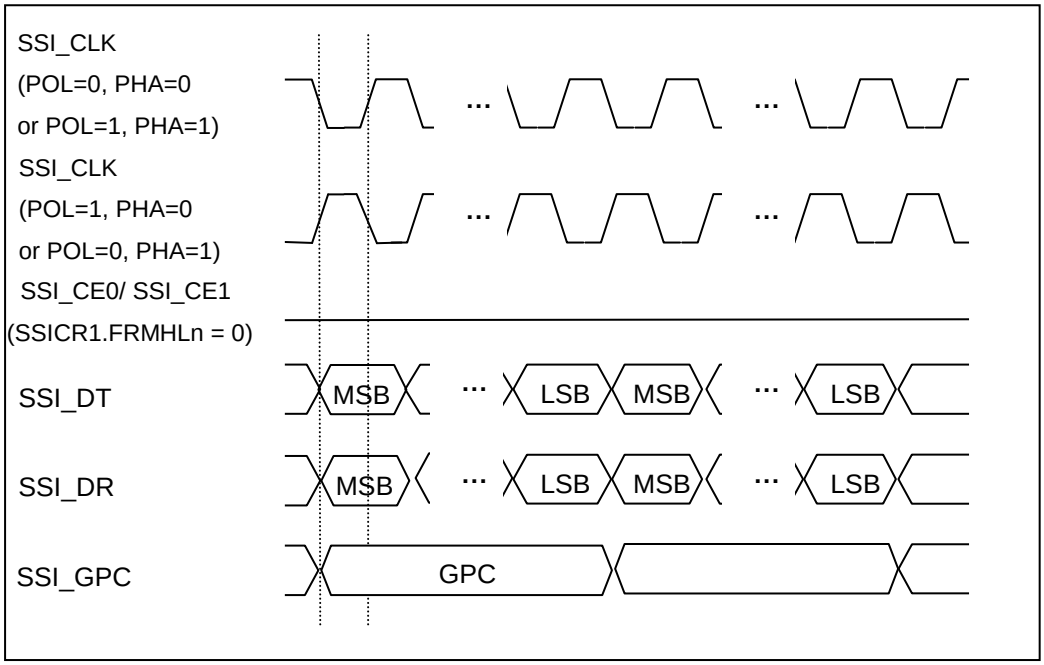


Figure 22-4 SPI Back-to-Back Transfer Format

For Motorola's SPI format transfers those continuous characters are exchanged during SSI_CE0 / SSI_CE1 being valid, the timing is illustrated in the figure (SSICR1.LFST = 0).

Back-to-back transfer is performed as transmit-only/full-duplex operation when transmit-FIFO is not empty before the completion of the last character's transfer or performed as receive-only operation.

22.6.1.3 Frame Interval Mode Transfer Format

When in interval mode (SSIITR. IVLTM ≠ '0'), SSI always wait for an interval time (SSIITR.IVLTM), transfer fixed number of characters (SSIICR), then repeats the operation.

When SSICR0.RFINE = 1, if transmit-FIFO is still empty after the interval time, receive-only transfer will occur.

During interval-wait time, SSI stops SSI_CLK, and when SSICR1.ITFRM = 0 it negates the SSI_CE0 / SSI_CE1, when SSICR1.ITFRM = 1 it keeps asserting the SSI_CE0 / SSI_CE1.

For transfers finished with transmit-FIFO empty, if the SSI transmit-FIFO is empty before fixed number of characters being loaded to transfer (SSICR1.UNFIN must be 1), then the SSI will set SSISR.UNDR = 1; if enabled, it'll send out a SSI under-run interrupt. At the same time, SSI will hold the SSI_CE0 / SSI_CE1 and SSI_CLK signals at current status and wait for the transmit-FIFO filling. The SSI will continue transfer after transmit-FIFO being filled. The SSI always stops after completion of fixed number of characters' transfer (SSICR1.UNFIN must be 0) with transmit-FIFO empty.

For transfers finished by SSICR0.RFINC being valid set, the SSI will stop after finished current character transfer and needn't wait for a whole completion of fixed number of characters' transfer.

Two Interval transfer mode are illustrated in the following figures. In these timing diagram, SSICR1.PHA = 0, SSICR1.POL = 0 and SSIICR = 0.

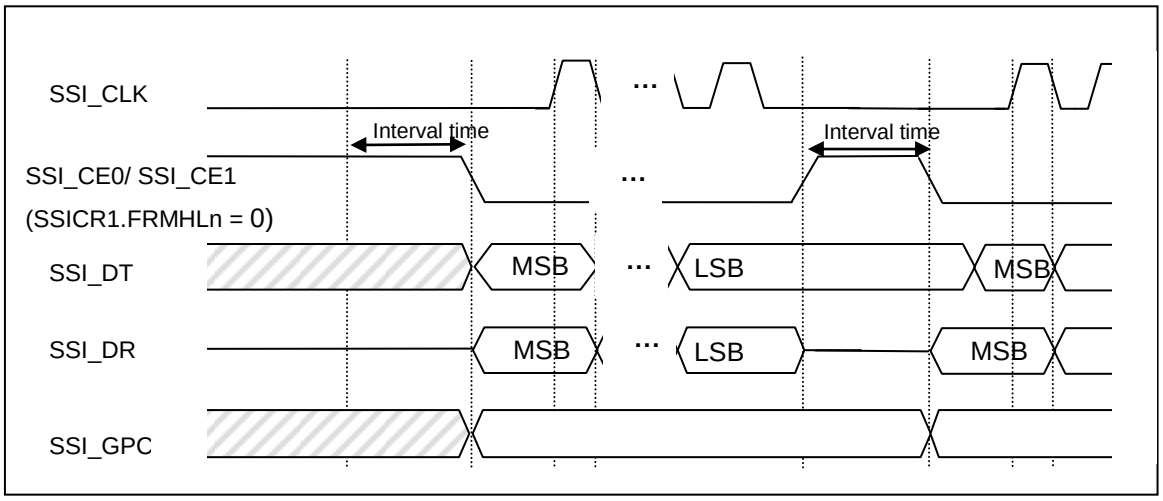


Figure 22-5 SPI Frame Interval Mode Transfer Format (ITFRM = 0, LFST = 0)

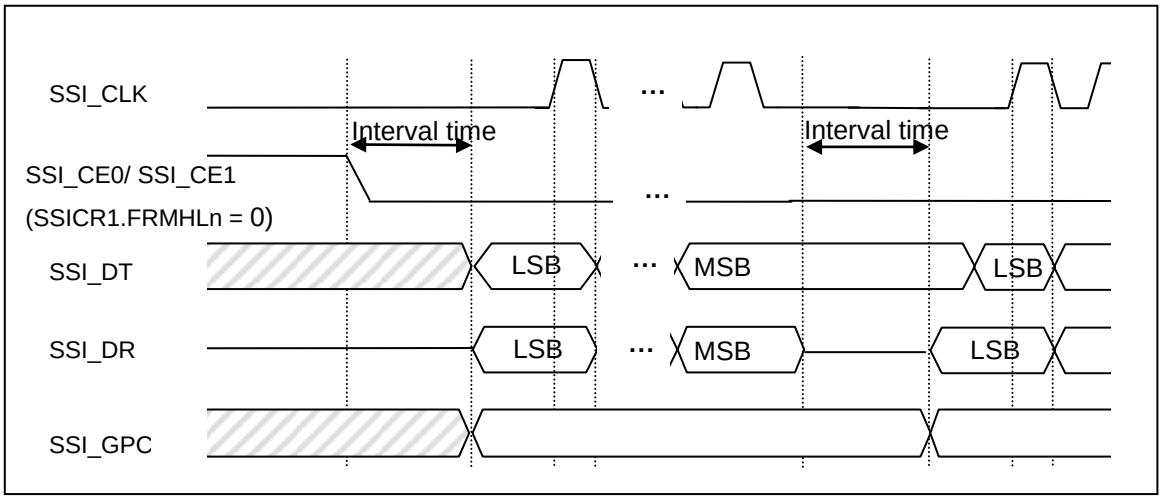


Figure 22-6 SPI Frame Interval Mode Transfer Format (ITFRM = 1, LFST = 1)

22.6.2 TI's SSP Format Details

In this format, each transfer begins with SSI_CE0 pulsed high for one SSI_CLK period. Then both master and slave drive data at SSI_CLK's rising edge and sample data at the falling edge. Data are transferred with MSB first or LSB first. At the end of the transfer, SSI_DT retains the value of the last bit sent through the next idle period.

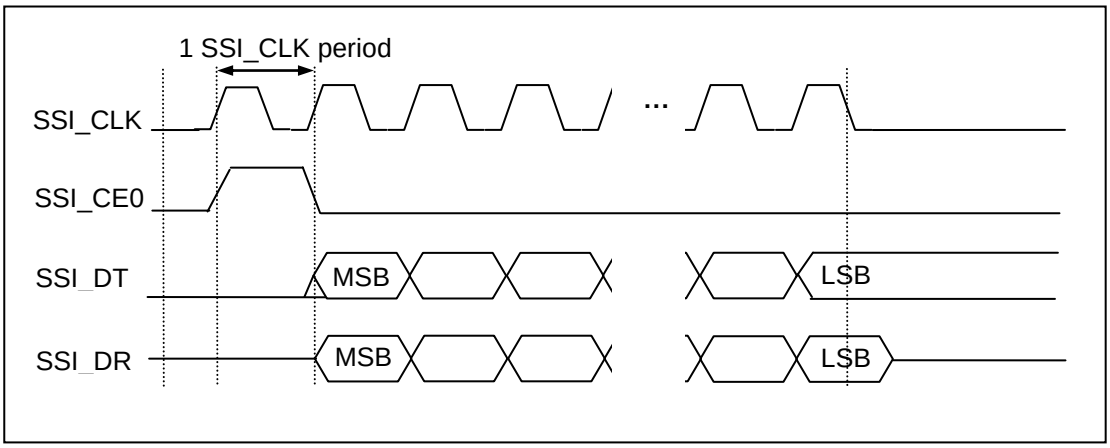


Figure 22-7 TI's SSP Single Transfer Format

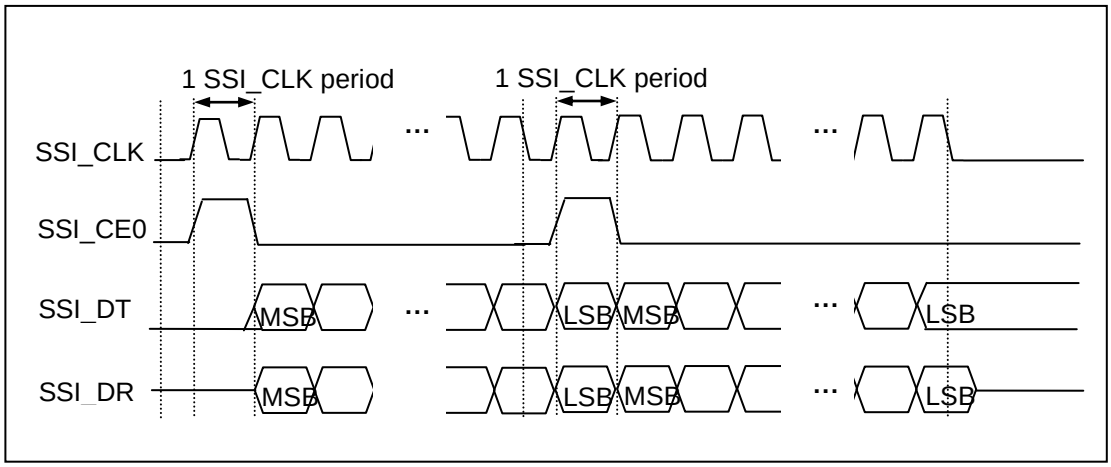


Figure 22-8 TI's SSP Back-to-back Transfer Format

22.6.3 National Microwire Format Details

It supports format 1 and format 2.

Format		Drive Edge	Sample Edge
NM1	Master	↓	↑
	Slave	↓	↑
NM2	Master	↓	↓
	Slave	↑	↑

If format 1 is selected, both master and slave drive data at SSI_CLK falling edge and sample data at the rising edge.

If format 2 is selected, master drive and sample data at SSI_CLK falling edge, slave drive and sample data at SSI_CLK rising edge.

SSI_CLK goes high midway through the command's MSB (or LSB) and continues to toggle at the bit rate. One bit clock (format 1) or half bit clock (format 2) period after the last command bit, the external slave must return the serial data requested, with MSB first (or LSB first) on SSI_DR.

SSI_CE0/SSI_CE2 de-asserts high half clock (SSI_CLK) period (and 1/2/3 additional clock periods) later.

Format 1 support back-to-back transfer, the start and end of back-to-back transfers are similar to those of a single transfer. However, SSI_CE0 / SSI_CE2 remains asserted throughout the transfer. The end of a character data on SSI_DR is immediately followed by the start of the next command byte on SSI_DT.

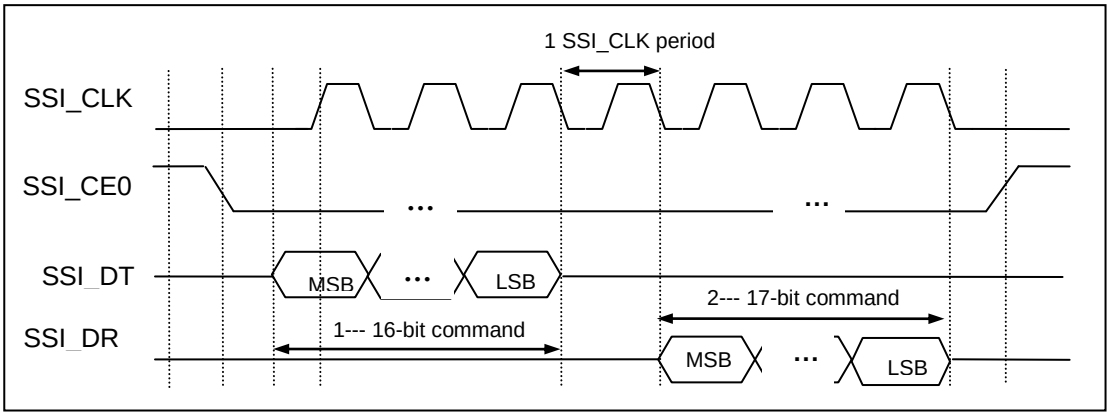


Figure 22-9 National Microwire Format 1 Single Transfer

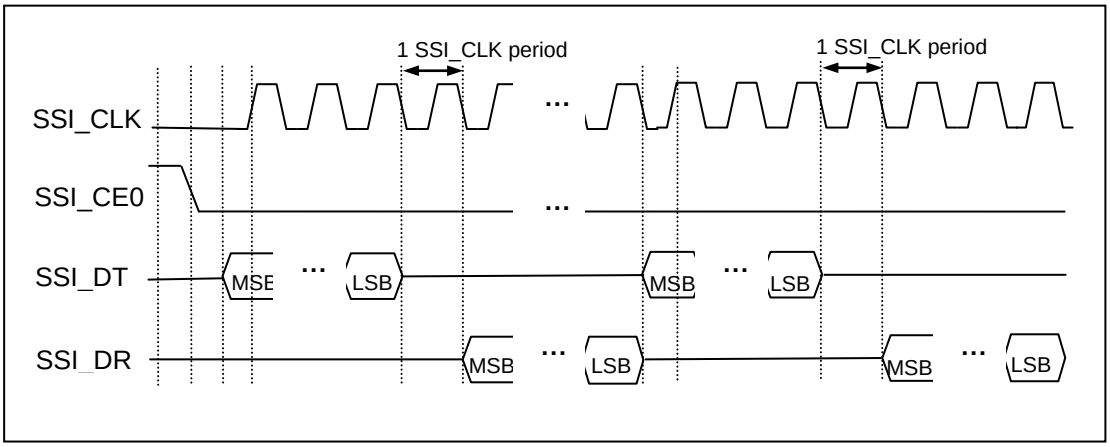


Figure 22-10 National Microwire Format 1 Back-to-back Transfer

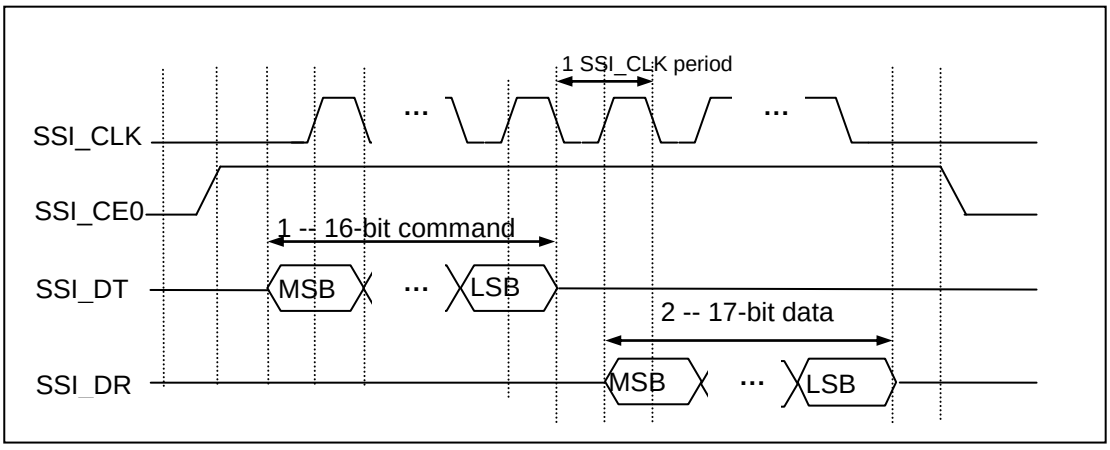


Figure 22-11 National Microwire Format 2 Read Timing

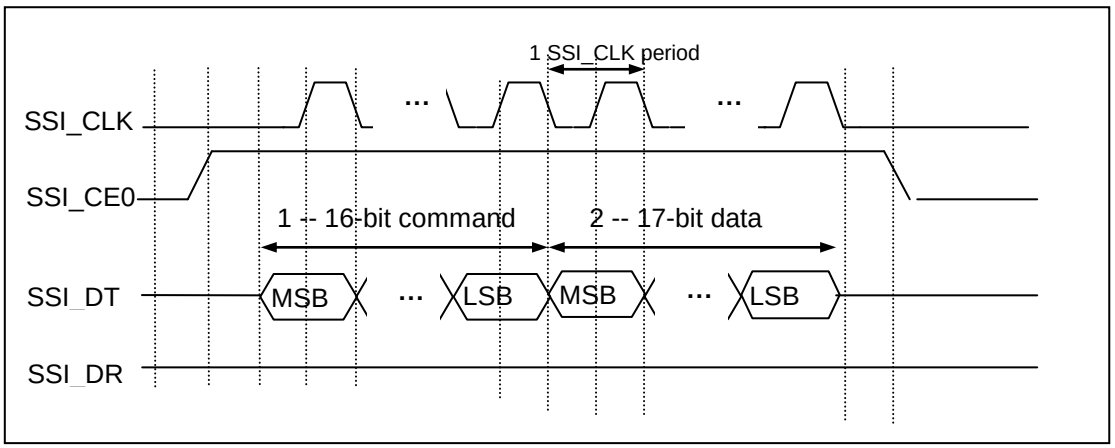


Figure 22-12 National Microwire Format 2 Write Timing

22.7 Register Description

Following chapter will descript the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. The registers can be read and written by APB bus
3. Register read/write attribute

   R - Read only

   W - Write only

   RW - read and write

   RCW - read and write, but clear to 0 by read

   RSW - read and write, but set to 1 by read

   RWC - read and write, clear to 0 by write 1, write 0 has no effect

   RWS - read and write, set to 1 by write 1, write 0 has no effect

   RC - read only, and clear to 0 by read

   RS - read only, and set to 1 by read

   SPEC - special access method, relate to its description
4. Reset Value

   1 - reset to 1

   0 - reset to 0

   ? - value unknown after reset

22.7.1 Register Mapping

The SSI's registers map base address is 0x1004_3000

The SSI has the following registers: one data, two control, one status, one bit-rate control, and two interval control registers. The table lists these registers.

Table 22-2 SSI Serial Port Registers

Name	Description	Reset Value	Address Offset	Access Size
SSI.SSIDR	Date register	0x????_????	0x0000	32
SSI.SSICR0	Controller register 0	0x0000_0000	0x0004	32
SSI.SSICR1	Controller register 1	0x0008_7860	0x0008	32
SSI.SSISR	Status register	0x0000_0098	0x000C	32
SSI.SSIITR	Interval mode controller register	0x????_0000	0x0010	16
SSI.SSIICR	Interval character-per-frame controller register	0x????_??00	0x0014	8
SSI.SSIGR	Bit-rate controller register	0x????_0000	0x0018	16
SSI.SSIRCNT	Receive counter register	0x0000_0000	0x001C	32

22.7.2 SSI Data Register (SSIDR)

	SSIDR																															Base + 0x0000							
Bit	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								
	D31-D17															GPC/D16	D15-D0																								
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?								

Bits	Name	Description	RW
31:17	D31-D17	D31-D17 to be written to/read from Transmit/Receive FIFO.	R
16	GPC/D16	This bit can be used as normal data bus bit 16 or GPC bit alternatively. When it is used as normal data bus bit, it's readable / writable; when SSI_GPC is used, it is GPC bit for SSI_GPC pin output and it's write-only.	RW
15:0	D15-D0	D15-D0 to be written to/read from Transmit/Receive FIFO. When the transfer frame length is less than 17-bit, received data is automatically right justified in the receive-FIFO and the upper unused bits are filled with '0'. For transmission, the upper unused bits of the data written into SSIDR is ignored by the transmit logic. (**NOTE:** "upper unused bits" does not include the SSIDR.GPC bit. National microwire format includes format 1 and format2, when national microwire format 2 is selected, Bit 16 of SSIDR is defined as read/write operation judge bit, if it is 0, bit 15~0 represent one read command; if it is 1, bit 15~0 represent one write command and following is the written data. So the maximum length of one command (is defined in MCOM) is 16, the maximum length of one written or read data (is defined in FLEN) can be 17. Transmit-FIFO only contain one read operation command once, or one	RW

		write operation command and its data once, after transmit-FIFO is empty, next command can be filled in transmit-FIFO.	

22.7.3 SSI Control Register0 (SSICR0)

	SSICR0																Base + 0x0004													
Bit	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
	Reserved												TENDIAN		RENDIAN		SSIE	TIE	RIE	TEIE	REIE	LOOP	RFINE	RFINC	EACLRUN	FSEL	Reserved	VRCNT	TFMODE	TFLUSH	RFLUSH	DISREV
RST	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

Bits	Name	Description	RW
31:20	Reserved	Writing has no effect, read as zero.	R
19:18	TENDIAN	Transfer endian 0: Most significant byte in a word first, most significant bit in a byte first. 1: Most significant byte in a word first, least significant bit in a byte first. 2: Least significant byte in a word first, least significant bit in a byte first. 3: Least significant byte in a word first, most significant bit in a byte first.	
17:16	RENDIAN	Receive endian 0: Most significant byte in a word first, most significant bit in a byte first. 1: Most significant byte in a word first, least significant bit in a byte first. 2: Least significant byte in a word first, least significant bit in a byte first. 3: Least significant byte in a word first, most significant bit in a byte first	RW
15	SSIE	This bit is used to enable/disable SSI module. 0: disable; 1: enable. Clearing SSIE will not reset SSI FIFO, SSICR0, SSICR1, SSIGR, SSIITR and SSIICR automatically. Software should ensure the FIFOs/registers are properly configured and be flush/reset manually when necessary before enabling SSI.	RW
14	TIE	This bit enables/disables the transmit-FIFO half-empty interrupt TXI. 0: disable; 1: enable.	RW
13	RIE	This bit enables/disables the receive-FIFO half-full interrupt RXI. 0: disable; 1: enable.	RW
12	TEIE	This bit enables/disables the transmit-error interrupt TEI. 0: disable; 1: enable.	RW
11	REIE	This bit enables/disables the receive-error interrupt REI. 0: disable; 1: enable.	RW
10	LOOP	Used for test purpose. In loop mode, the output of SSI transmit shift register is connected to input of SSI receive shift register internally. The data received should be the same as the data transmitted. And do not output any valid signals on the pins. 0: normal SSI mode; 1: LOOP mode.	RW

<table>
<tr><td>9</td><td>RFINE</td><td>Receive finish control enable.
0: disable; 1: enable.
For SSICR1.FMAT = B’10 (National Microwire format 1 is selected), SSICR0.RFINE must be 0.</td><td rowspan="2">The receive finish condition list below:<table><tr><th>RFINE</th><th>RFINC</th><th>Receive Finish Condition</th></tr><tr><td>0</td><td>x</td><td>Same as transmit completion condition (transmit-fifo is empty and SSICR1.UNFIN = 0)</td></tr><tr><td>1</td><td>0</td><td>Receive continue</td></tr><tr><td>1</td><td>1</td><td>Receive finish</td></tr></table></td><td>RW</td></tr>
<tr><td>8</td><td>RFINC*</td><td>Receive finish control bit.
0: receive continue
1: receive finished</td><td>RW</td></tr>
<tr><td>7</td><td>EACLRUN</td><td colspan="2">0: don’t auto clear under flag, software clear under
1: software auto clear under flag when tfifo don’t empty</td><td>RW</td></tr>
<tr><td>6</td><td>FSEL</td><td colspan="2">This bit sets the frame signal to be used for slave select. The unselected frame signal always output invalid level.
0: SSI_CE0 is selected
1: SSI_CE1 is selected</td><td>RW</td></tr>
<tr><td>5</td><td>Reserved</td><td colspan="2">Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>4</td><td>VRCNT</td><td colspan="2">Receive counter (RCNT) valid flag.
0: SSIRCNT.RCNT will be ignored.
1: SSIRCNT.RCNT will be used.</td><td>RW</td></tr>
<tr><td>3</td><td>TFMODE</td><td colspan="2">0: new fifo empty mode
1: old fifo empty mode</td><td>RW</td></tr>
<tr><td>2</td><td>TFLUSH</td><td colspan="2">Flush the transmit FIFO when set to 1.
Always return 0 when read.</td><td>RW</td></tr>
<tr><td>1</td><td>RFLUSH</td><td colspan="2">Flush the receive FIFO when set to 1.
Always return 0 when read.</td><td>RW</td></tr>
<tr><td>0</td><td>DISREV</td><td colspan="2">Receive function enable.
0: enable.
1: disable.</td><td>RW</td></tr>
</table>

NOTE:

***:** When transmitting finished or for receive-only operation, transmit function can be disabled and this bit is used to control receiving completion, and the SSI will consume less power.

When the finish condition is set, the receiving will complete after present character is completely shifted in, then the SSI will stop the SSI_CLK and negate the SSI_CE0 / SSI_CE1 if necessary. To make sure present transfer is completed, user must read and get SSISR.END = 1 (or SSISR.BUSY = 0).

22.7.4 SSI Control Register1 (SSICR1)

	SSICR1																Base + 0x0008													
Bit	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
	FRMHL		TFVCK		TCKFI		Reserved	ITFRM	UNFIN	Reserved	FMAT		TTRG				MCOM				RTRG				FLEN					Reserved	PHA	POL
RST	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	0	0	0	0

Bits	Name	Description	RW
31:30	FRMHL	Frame valid level select, FRMHL [1: 0] correspond to SSI_CE1 and SSI_CE0 respectively. **FRMHL[1:0]** / **Description** / 00 / SSI_CE0 is low level valid and SSI_CE1 is low level valid / Initial value 01 / SSI_CE0 is high level valid and SSI_CE1 is low level valid / 10 / SSI_CE0 is low level valid and SSI_CE1 is high level valid / 11 / SSI_CE0 is high level valid and SSI_CE1 is high level valid /	RW
29:28	TFVCK	Time from frame valid to clock start. When TFVCK = B'00, the time is fixed half or one SSI_CLK cycle according to SSICR1.POL and SSICR1.PHA configuration. For SSICR1.FMAT = B'01, SSICR1.TFVCK is ignored. **TFVCK[1:0]** / **Description** / 00 / half or one SSI_CLK cycle delay time / Initial value 01 / 1 more SSI_CLK cycle delay is added / 10 / 2 more SSI_CLK cycle delay is added / 11 / 3 more SSI_CLK cycle delay is added /	RW
27:26	TCKFI	Time from clock stop to frame invalid. When TCKFI = B'00, the time is fixed one SSI_CLK or half SSI_CLK cycle according to SSICR1.POL and SSICR1.PHA configuration. For SSICR1.FMAT = B'01, SSICR1.TFVCK is ignored. **TCKFI[1:0]** / **Description** / 00 / half or one SSI_CLK cycle delay / Initial value 01 / 1 more SSI_CLK cycle delay is added / 10 / 2 more SSI_CLK cycle delay is added / 11 / 3 more SSI_CLK cycle delay is added /	RW
25	Reserved	Writing has no effect, read as zero.	R
24	ITFRM	Frame during interval, selects if the Frame (SSI_CE0 /SSI_CE1) signal is	RW

<table>
<tr><td></td><td></td><td>negated or not during interval time at Interval Mode (SSICR1.FMAT = B'00 and SSIITR.IVLTM ≠ H'0000). It's ignored at Normal Mode.
0: SSI_CE0 /SSI_CE1 de-asserts during interval time at Interval Mode
1: SSI_CE0 /SSI_CE1 keeps asserted during interval time at Interval Mode</td><td></td></tr>
<tr><td>23</td><td>UNFIN</td><td>This bit controls whether the SSI finishes transmission or wait for data filling (underrun happen) after all data in transmit-FIFO are sent out during transfer. This bit must be cleared to 0 when SSICR1.FMAT = B'01. (TI's SSP format)
0: TxFIFO empty means end of transmission
1: Transmission didn't finish when transmit-FIFO is empty, SSI underrun error would occur and SSI waits for data filling; SSI_CLK and SSI_CE0 /SSI_CE1 keeps asserted, SSI_CLK stop at the current level
NOTE: For TxFIFO empty before any transfer after SSI enabled, if SSICR1.UNFIN = 1 or SSICR0.RFINE = 0, SSI will wait till TxFIFO isn't empty then start to transfer and no underrun error will occur; if SSICR1.UNFIN = 0 and SSICR0.RFINE = 1, after transmit-FIFO become empty, SSI will start a receive-only transfer.</td><td>RW</td></tr>
<tr><td>22</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>21:20</td><td>FMAT</td><td>These bits set the operating transfer format.
<table>
<tr><th>FMAT</th><th>Description</th><th></th></tr>
<tr><td>00</td><td>Motorola's SPI format</td><td>Initial value</td></tr>
<tr><td>01</td><td>TI's SSP format</td><td></td></tr>
<tr><td>10</td><td>National Microwire 1 format</td><td></td></tr>
<tr><td>11</td><td>National Microwire 2 format</td><td></td></tr>
</table></td><td>RW</td></tr>
<tr><td>19:16</td><td>TTRG</td><td>These bits define the transmit-FIFO half-empty threshold value, which when equal or less characters left in transmit-FIFO, the SSISR.TFHE will be set to '1'.
0000: less than or equal to 1
n: less than or equal to nx8</td><td>RW</td></tr>
<tr><td>15:12</td><td>MCOM</td><td>Defines the length of command.
Only used when SSICR1.FMAT = b'10 or b'11 (National Microwire format 1 or 2 is selected).
The command length is (MCOM + 1).
Initial value is b'0111.</td><td>RW</td></tr>
<tr><td>11:8</td><td>RTRG</td><td>These bits define the receive-FIFO half-full threshold value, which when equal or more characters received in receive-FIFO, the SSISR.RFHF will be set to '1'.
0000: more than or equal to 1
n: more than or equal to nx8</td><td>RW</td></tr>
<tr><td>7:3</td><td>FLEN</td><td>These bits set the bit length of every character to be transmitted/received. The maximum data length can be configured is 32 bits. For data length</td><td>RW</td></tr>
</table>

		longer than 17 bits (multiples of the SSICR1.FLEN configured length), the software should ensure properly processing. When SSI_GPC pin is used, the FLEN shouldn't be configured as B'01111 (17-bit data). When TI SSP mode is selected (FMAT = 2'b01), 2-bit data length (FLEN = 0) isn't supported. The bit length of one data = FLEN + 2	
2	Reserved	Writing has no effect, read as zero.	R
1	PHA	This bit sets the phase of the SSI_CLK from the beginning of a data frame for Motorola's SPI format (SSICR1.FMAT = B'00). 0: The leading edge of SSI_CLK is used to sample data from SSI_DR after the SSI_CE0 /SSI_CE1 goes valid, it is initial value 1: The leading edge of SSI_CLK is used to drive data onto SSI_DT after the SSI_CE0 /SSI_CE1 goes valid	RW
0	POL	SSI_CLK polarity for Motorola's SPI format during idle state. (SSICR1.FMAT = B'00). 0: SSI_CLK keeps low level when idle 1: SSI_CLK keeps high level when idle	RW

22.7.5 SSI Status Register (SSISR)

	SSISR																Base + 0x000C													
Bit	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
	Reserved								TFIFO-NUM								RFIFO-NUM								END	BUSY	TFF	RFE	TFHE	RFHF	UNDR	OVER
RST	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	0	0	1	1	0	0	0

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	TFIFO-NUM	These bits indicate the Characters Number in Transmit-FIFO.	R
15:8	RFIFO-NUM	These bits indicate the Characters Number in Transmit-FIFO.	R
7	END	This bit indicates transfer end status. It is the inverse of SSISR.BUSY when transfer is in process, but it'll keep cleared at interval time before transfer is completed. It'll be set when transfer finished.	R
6	BUSY	This bit indicates SSI's working status. 0: SSI is idle or at interval time 1: Transmission and/or reception is in process	R
5	TFF	This bit denotes transmit-FIFO is full or not. 0: Transmit-FIFO is not full 1: Transmit-FIFO is full	R

4	RFE	This bit denotes receive-FIFO is empty or not. 0: Receive-FIFO is not empty 1: Receive-FIFO is empty	R
3	TFHE	This bit denotes whether the characters number in transmit-FIFO being less or equal to SSICR1.TTRG. 0: The data in transmit-FIFO is more than the condition set by SSICR1.TTRG 1: The data in transmit-FIFO meets the condition set by SSICR1.TTRG, If SSICR0.TIE = 1, it will generate SSI TXI interrupt	R
2	RFHF	This bit denotes whether the characters number in receive-FIFO being more or equal to the number set by SSICR1.RTRG. 0: The data in receive-FIFO is less than the condition set by SSICR1.RTRG 1: The data in receive-FIFO meets the condition set by SSICR1.RTRG, If SSICR0.RIE = 1, it will generate SSI RXI interrupt	R
1	UNDR	Transmit-FIFO underrun status. When underrun happens, SSI set this bit and keeps the current status of SSI_CLK and SSI_CE0/SSI_CE1, waiting for transmit-FIFO filling. 0: Underrun has not occurred 1: Underrun has occurred, when SSICR0.TEIE is set, it will generate SSI TEI interrupt. Write '0' to clear this bit, writing '1' has no effect	RW
0	OVER	Receive-FIFO overrun status, new received data will lose. 0: Overrun has not occurred 1: Overrun has occurred; When SSICR0.REIE is set, it will generate SSI REI interrupt. Write '0' to clear this bit, writing '1' has no effect	RW

22.7.6 SSI Interval Time Control Register (SSIITR)

	SSIITR																Base + 0x0010															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	CNTCLK	IVLTM														
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15	CNTCLK	Counting clock source select. 0: Use SSI bit clock (SSI_CLK) as the interval counter clock source 1: Use 32K clock as the interval counter clock source	RW

14:0	IVLTM	Interval time set, set the cycle number of counting clock source for desired interval time. When SSIITR.IVLTM = 0x0000, normal mode is selected, and SSIITR.CNTCLK and SSIICR are ignored. When SSIITR.IVLTM ≠ 0x0000, interval mode is selected. The interval time is calculated as follows: *Interval time ≈ [CNTCLK clock period] * [Value of IVLTM]* The actual interval time is as follow: When SSIITR.CNTCLK = 0: *Interval time = [CNTCLK clock period] * [Value of IVLTM] + 3 * device_clock period* When SSIITR.CNTCLK = 1: *Interval time ≥ [CNTCLK clock period] * [Value of IVLTM + 1] + 1 * device_clock period;* *Interval time ≤ [CNTCLK clock period] * [Value of IVLTM + 2] + 2 * device_clock period*	RW

22.7.7 SSI Interval Character-per-frame Control Register (SSIICR)

	SSIICR																Base + 0x0014											
Bit																							7	6	5	4	3	2	1	0
	Reserved																											ICC		
RST																							0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:3	Reserved	Writing has no effect, read as zero.	R
2:0	ICC	Sets the fixed number of characters to be transmitted / received each time during SSI_CLK changing (and SSI_CE0 / SSI_CE1 asserting) in interval mode for SSICR1.FMAT = B'00 (Motorola's SPI format is selected). SSIICR is ignored for SSICR1.FMAT ≠ B'00. The desired transfer number of characters-per-frame is (SSIICR set value + 1).	RW

22.7.8 SSI Clock Generator Register (SSIGR)

	SSIGR																Base + 0x0018															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																								CGV							
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:8	Reserved	Writing has no effect, read as zero.	R
7:0	CGV	Sets the frequency of serial bit clock (SSI_CLK). The serial bit clock (SSI_CLK) is generated by dividing device-clock as follows: F_{SSI_CLK} = [Frequency of device clock] / (2 * (CGV + 1)) Device clock is generated in CPM module. The value in SSIGR can be set from 0 to 255, and initialized to 0x0000 on power-on reset.	RW

22.7.9 SSI Receive Counter Register (SSIRCNT)

	SSIRCNT																Base + 0x001C															
Bit	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
	Reserved																RCNT															
RST	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

Bits	Name	Description	RW
31:16	Reserved	Writing has no effect, read as zero.	R
15:0	RCNT	Number of data to be received. If SSICR1.UNFIN = 0 and SSICR0.RFINE = 1, after transmit-FIFO become empty, SSI will start a receive-only transfer. If SSICR0.VRCNT = 1, a special internal counter will be increased automatically. When the value of the internal counter equals to the RCNT, the receiving operation will be stopped automatically.	RW

22.8 Software Guideline

22.8.1 Common flow

The initial steps are:

1) set GPIO
2) disable interrupt (or not) by setting SSICR0
3) clear SSISR (write 0x0, clear interrupt, FIFO state and working condition)
4) flush FIFO (write SSICR0.xFLUSH once for each FIFO)
5) select the counting clock by configuring SSIITR
6) configure SSIGR (set the frequency of serial bit clock)
7) configure SSICR1 (set valid level of CE signals, set head and end delay of the transfer, set state of CE signals in interval time, select protocols, set threshold of FIFO, set bit length, etc.)
8) configure SSIICR (set the fixed number of characters to be transmit/received each time during SSI_CLK changing in interval mode)
9) configure SSIRCNT (number of data to be received)
10) configure SSICR0 (the start trigger is in this register)

NOTE: the frequency of dev_clk should not be 4 times or more higher than PCLK

22.8.2 Interrupt Operation

In SSI, there are TXI, RXI, TEI and REI total 4 interrupts, all these interrupts are combined together to make one SSI interrupt, which can be masked by writing '1' into corresponding mask bit in INTC interrupt mask register (IMR).

Table 22-3 SSI Interrupts

Operation	Condition	Flag Bit	Mask Bit	Interrupt	DMAC Activation
Transmit	T-FIFO is half-empty or less	SSISR.TFHE	SSICR0.TIE	**TXI**	Possible
	Transmit underrun error	SSISR.UNDR	SSICR0.TEIE	**TEI**	Impossible
Receive	R-FIFO is half-full or more	SSISR.RFHF	SSICR0.RIE	**RXI**	Possible
	Receive overrun error	SSISR.OVER	SSICR0.REIE	**REI**	Impossible

Either SSISR.TFHE or SSISR.RFHF can activate DMA transferring when corresponding individual interrupt mask bit in SSICR0 is cleared (masked) and DMA is enabled and configured.

22.9 Index

Name	Description
APB	Advanced Microcontroller Bus Architecture
Dual SPI	See SPI
CPU	Central Processing Unit
Endian	The convention used to interpret the bytes making up a data word when those bytes are stored in computer memory. See Endianness at Wikipedia
FIFO	First In, First Out, a method for organizing and manipulating a data buffer

LSB	See Endian
MSB	See Endian
Quad SPI	See SPI
SPI	Serial Peripheral Interface
Standard SPI	See SPI

23 Universal Asynchronous Receiver/Transmitter(uart)

23.1 Overview

This chapter describes the universal asynchronous receiver/transmitter (UART) serial ports. There are three UARTs: All UARTs use the same programming model. Each of the serial ports can operate in interrupt-based mode or DMA-based mode.

The Universal asynchronous receiver/transmitter (UART) is compatible with the 16550-industry standard and can be used as slow infrared asynchronous interface that conforms to the Infrared Data Association (IrDA) serial infrared specification 1.1.

23.2 Features

- Full-duplex operation
- 5-, 6-, 7- or 8-bit characters with optional no parity or even or odd parity and with 1, 1½, or 2 stop bits
- 64x8 bit transmit FIFO and 64x11bit receive FIFO
- Transmission, reception and line status Independently
- Internal diagnostic capability Loopback control and break, parity, overrun and framing-error is provided
- Separate DMA requests for transmit and receive data services in FIFO mode
- Supports modem flow control by software or hardware
- Slow infrared asynchronous interface that conforms to IrDA specification

23.3 Block Diagram

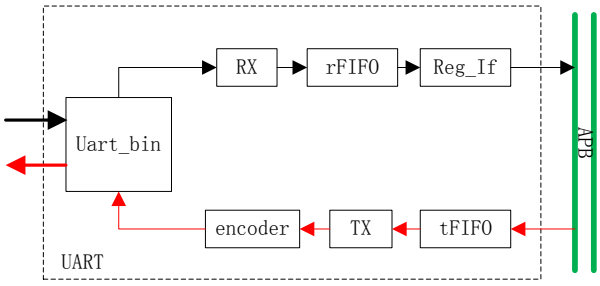


Figure 23-1 UART DATA Controller Block Diagram

As shown in UART DATA Controller Block Diagram，the two part includes receiving data and sending data in UART; From PIN to receive the data processing and put it on the APB bus. For the transmission direction,UART gets the data from the APB bus and sent to the PIN.

23.4 Functional Description

23.4.1 Full-duplex operation

UART bidirectional communication，can realize full duplex transmission and reception.

23.4.2 The meaning of all bits

1. start : first sends a signal to a logic "0", to indicate the beginning of transmissiong or reception character.
2. data bit : after the start bit. data bits can be 5,6,7,8etc, constitute a character. Began to transfer from the lowest.
3. parity bit : it is optional and used to check the accuracy of data transmission and reception.
4. stop bit : it marks the end of a character data. can be high level 1,1.5,2. stop bit is not only the end of said tranmission,and provides computer correction clock synchronization opportunity.
5. idle bit : In a logic "1" state, said the current line without data transfer.

23.4.3 RFIFO and TFIFO

RFIFO is 64*8bit receiving FIFO; TFIFO is 64*11bit tranmission FIFO. they used to cache data.
The data in the UART registers(temporary memory block), and then through the FIFO transmitted to the serial device, without the FIFO, the information will become be, may not transfer to Modem.

23.4.4 Transmission, reception and line status Independently

Independently controlled transmit, receive (data ready or timeout), line status interrupts

23.4.5 Slow infrared asynchronous interface

UART also contains a IrDA serial infrared(SIR) encorder/decoder module. IrDA SIR module is used to convert between asynchronous UART data stream and a half duplex serial SIR interface. The SIR module's task is to give the UART to provide a digital output code and a decoding input . UART signal pin and a infrared transceiver connected to implement IrDA SIR physical layer connection.

23.5 Pins Description

Table 23-1 UART Pins Description

Name	Type	Description
RxD	Input	Receive data input
TxD	Output	Transmit data output
CTS_	Input	Clear to Send --- Modem Transmission enabled
RTS_	Output	Request to Send --- UART Transmission request

NOTE: UART2, UART1, UART0 support RxD, TxD, RTS_, CTS_.

23.6 Data Format Description

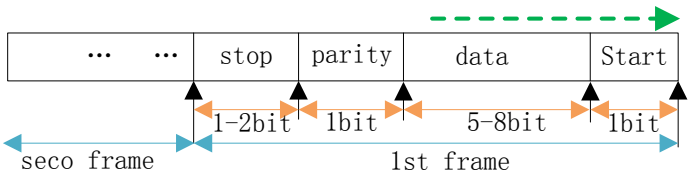


Figure 23-2 UART Frame Structure Diagram

As shown in UART Frame Structure Diagram, the frame data from left to right transmission. The start bit and parity bit are all 1bit; Data bits and stop bits can be changed, so data bits range is 5bit to 8bit; stop bits range is 1bit ,1.5bit and 2bit.

23.7 Register Description

Following chapter will describe the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute
   - R- Read only
   - W- Write only
   - RW- Read and Write
   - RCW- Read and Write, but Clear to 0 by Read
   - RSW- Read and Write, but Set to 1 by Read
   - RWC- Read and Write, Clear to 0 by Write 1, Write 0 has no effect
   - RWS- Read and Write, Set to 1 by Write 1, Write 0 has no effect
   - RC- Read only, and Clear to 0 by read
   - RS- Read only, and Set to 1 by read
3. Reset Value
   - 1 - reset to 1
   - 0 - reset to 0
   - ? - value unknown after reset

23.7.1 Register Memory Map

There are 3 UART controllers in chip, name UART0, UART1, UART2 respectively. The base address of UARTS(s) as follows:

Table 23-2 Registers Memory Map Base Address

Name	Base	Description
UART0	0x10030000	Address base of UART0
UART1	0x10031000	Address base of UART1
UART2	0x10032000	Address base of UART2

To describe all UART's registers as shown in Table 23-3：

Table 23-3 UART Registers Map

Name	RW	Reset Value	Address Offset	Access Size
URBR	R	0x??	0x000	8
UTHR	W	0x??	0x000	8
UDLLR	RW	0x00	0x000	8
UDLHR	RW	0x00	0x004	8
UIER	RW	0x00	0x004	8
UIIR	R	0x01	0x008	8
UFCR	W	0x00	0x008	8
ULCR	RW	0x00	0x00c	8
UMCR	RW	0x00	0x010	8
ULSR	R	0x60	0x014	8
UMSR	R	0x00	0x018	8
USPR	RW	0x00	0x01c	8
ISR	RW	0x00	0x020	8
UMR	RW	0x00	0x024	8
UACR	RW	0x00	0x028	16
URCR	R	0x00	0x040	8
UTCR	R	0x00	0x044	8

23.7.2 Register and Fields Description

23.7.2.1 UART Receive Buffer Register(URBR,0x0000)

The read-only URBR is corresponded to one level 11bit buffer in non-FIFO mode and a 32x11bit FIFO that holds the character(s) received by the UART. Bits in URBR are right justified when being configured to use fewer than eight bits, and the rest of most significant data bits are zeroed and the most significant three bits of each buffer are the status for the character in the buffer. If ULSR.DRY is 0, don't read URBR, otherwise wrong operation may occur。

	URBR																BASE + 0x0000													
Bit	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
	Reserved																								URBR							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?

Bits	Name	Description	RW

31:8	Reserved	Writing has no effect. Read as zero.	R
7:0	URBR	8-bit UART receive read data.	R

GNOTE:

The URBR register is valid when ULCR register's field DLAB = 0 and cpu to request read operation;

23.7.2.2 UART Transmit Hold Register (UTHR,0x0000)

The write-only UTHR is corresponded to one level 8 bit buffer in non-FIFO mode and a 32x8bit FIFO in FIFO mode that holds the data byte(s) to be transmitted next.

	UTHR																BASE + 0x0000													
Bit	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
	Reserved																								UTHR							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect. Read as zero.	R
7:0	UTHR	8-bit UART transmit write hold data.	W

GNOTE:

The UTHR register is valid when ULCR register's field DLAB = 0 and cpu to request writing operation;

23.7.2.3 UART Divisor Latch Low Register (UDLLR,0x000)

UART Divisor Latch registers, UDLLR/UDLHR together compose the divisor for the programmable baud rate generator that can take the UART device clock and divide it by 1 to (2^{16} – 1).

The UART device source clock is EXCLK or EXCLK/2 that is determined by CPCCR.ECS.

UDLHR/UDLLR stores the high/low 8-bit of the divisor respectively. Load these divisor latches during initialization to ensure that the baud rate generator operates properly. If both Divisor Latch registers are 0, the 16X clock stops.

If you don't set UMR and UACR, UART will work at normal mode with the specified frequency. The relationship between baud rate and the value of Divisor is shown by the formula when UMR and UACR are not set:

Baud Rate = (UART device clock) / (16 * **Divisor**)

	UDLLR																													BASE + 0x000							
Bit	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
	Reserved																								Divisor Latch Low 8-bit							
RST	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

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect. Read as zero.	R
7:0	DLL_8bit	Low 8-bit divisor Latch	W

GNOTE:

The UDLLR register is valid when ULCR register's field DLAB = 1;

23.7.2.4 UART Divisor Latch High Register (UDLHR,0x004)

The interpretation of the same with UDLLR.

	UDLHR																													BASE + 0x004							
Bit	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
	Reserved																								Divisor Latch high 8-bit							
RST	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

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect. Read as zero.	R
7:0	DLH_8bit	high 8-bit divisor Latch	W

GNOTE:

The UDLHR register is valid when ULCR register's field DLAB = 1;

23.7.2.5 UART Interrupt Enable Register (UIER,0x004)

The UART Interrupt Enable Register (UIER) contains the interrupt enable bits for the five types of interrupts (receive data ready, timeout, line status, and transmit data request, and modem status) that set a value in UIIR.

	UIER																													BASE + 0x004							
Bit	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
	Reserved																								Reserved			RTOIE	MSIE	RLSIE	TDRIE	RDRIE
RST	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

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect. Read as zero.	R
7:5	Reserved	Writing has no effect, read as zero.	R
4	RTOIE	**Receive Timeout Interrupt Enable.** 0: Disable the receive timeout interrupt 1: Enable the receive timeout interrupt Timeout means the URDR (FIFO mode) is not empty but no character has received for a period of time T: T (bits) = 4 X Word length + 12.	RW
3	MSIE	**Modem Status Interrupt Enable.** 0: Disable the modem status interrupt 1: Enable the modem status interrupt	RW
2	RLSIE	**Receive Line Status Interrupt Enable.** 0: Disable receive line status interrupt 1: Enable receive line status interrupt	RW
1	TDRIE	**Transmit Data Request Interrupt Enable.** 0: Disable the transmit data request interrupt 1: Enable the transmit data request interrupt	RW
0	RDRIE	**Receive Data Ready Interrupt Enable.** 0: Disable the receive data ready interrupt 1: Enable the receive data ready interrupt	RW

GNOTE:The UIER register is valid when ULCR register's field DLAB = 0;

23.7.2.6 UART Interrupt Identification Register (UIIR,0x008)

The read-only UART Interrupt Identification Register (UIIR) records the prioritized pending interrupt source information. Its initial value after power-on reset is 0x01.

	UIIR																													BASE + 0x008							
Bit	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
	Reserved																								FFMSEL	Reserved			INID			INPEND
RST	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

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect. Read as zero.	R
7:6	FFMSEL	**FIFO Mode Select.** 0b00: Non-FIFO mode 0b01: Reserved 0b10: Reserved 0b11: FIFO mode	R
5:4	Reserved	Writing has no effect, read as zero.	R
3:1	INID	**Interrupt Identifier.** These bits identify the current highest priority pending interrupt. INID: 0b000 Modem Status; 0b001 Transmit Data Request; 0b010 Receive Data Ready; 0b011 Receive Line Status; 0b100 Reserved; 0b101 Reserved; 0b110 Receive Time Out; 0b111 Reserved See _ for details.	R
0	INPEND	**Interrupt Pending.** 0: interrupt is pending 1: No interrupt pending	R

Table 23-4 UART Interrupt Identification Register Description

UIIR.INID	Interrupt Set/Clear Cause			
	Priority	Type	Source	Clear Condition
0b0001	—	None	No pending interrupt	—
0b0110	1st Highest	Receive Line Status	Overrun, Parity, Frame Error, Break Interrupt, and FIFO Error (DMA mode only)	Reading ULSR or empty all the error characters in DMA mode
0b0100	2nd Highest	Receive Data Ready	FIFO mode: Trigger threshold was reached Non-FIFO mode: URBR full	FIFO mode: Reading URBR till below trigger threshold. Non-FIFO mode: Empty URBR

0b1100	2nd Highest	Receive Timeout	FIFO mode only: URBR not empty but no data read in for a period of time	Reset receive buffer by setting UFCR.RFRT to 1 or Reading URBR
0b0010	3rd Highest	Transmit Data Request	FIFO mode: Empty location in UTHR equal to half or more than half Non-FIFO mode: UTHR empty	FIFO mode: Data number in UTHR more than half Non-FIFO mode: Writing UTHR
0b0000	4th Highest	Modem Status	Modem CTS_ pin status change	Reading UMSR

23.7.2.7 UART FIFO Control Register (UFCR,0x008)

The write-only register UFCR contains the control bits for receive and transmit FIFO.

	UFCR																													BASE + 0x008
Bit	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
	Reserved																								RDTR		Reserved	UME	DME	TFRT	RFRT	FME
RST	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

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect. Read as zero.	R
7:6	RDTR	**Receive Buffer Data Number Trigger.** These bits are used to select the trigger level for the receive data ready interrupt in FIFO mode. 0b00: 1 0b01: 16 0b10: 32 0b11: 60	RW
5	Reserved	Writing has no effect, read as zero.	R
4	UME	**UART Module Enable.** 0: Disable UART 1: Enable UART	RW
3	DME	**DMA Mode Enable.** 0: Disable DMA mode 1: Enable DMA mode	RW
2	TFRT	**Transmit Holding Register Reset.** 0: Not reset 1: Reset transmit FIFO	RW

1	RFRT	**Receive Buffer Reset.** 0: Not reset 1: Reset receive FIFO	RW
0	FME	**FIFO Mode Enable.** Set this bit before the trigger levels. 0: non-FIFO mode 1: FIFO mode	RW

23.7.2.8 UART Line Control Register (ULCR,0x00C)

The ULCR defines the format for UART data transmission.

	ULCR																BASE + 0x00C													
Bit	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
	Reserved																								DLAB	SBK	STPAR	PARM	PARE	SBLS	WLS	
RST	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

Bits	Name	Description	RW
7	DLAB	**Divisor Latch Access Bit.** 0: Enable to access URBR, UTHR or UIER 1: Enable to access UDLLR or UDLHR	RW
6	SBK	**Set Break.** Causes a break condition (at least one 0x00 data) to be transmitted to the receiving UART. Acts only on the TXD pin and has no effect on the transmit logic. 0: No effect on TXD output 1: Forces TXD output to 0	RW
5	STPAR	**Sticky Parity.** Setting this bit forces parity location to be opposite of PARM bit when PARE is 1 (it is ignored when PARE is 0). 0: Disable Sticky parity 1: Enable Sticky parity (opposite o f PARM bit)	RW
4	PARM	**Parity Odd/Even Mode Select.** If PARE = 0, PARM is ignored. 0: Odd parity 1: Even parity	RW
3	PARE	**Parity Enable.** Enables a parity bit to be generated on transmission or checked on reception. 0: No parity	RW

		1: Parity	
2	SBLS	**Stop Bit Length Select.** Specifies the number of stop bits transmitted and received in each character. When receiving, the receiver checks only the first stop bit. 0: 1 stop bit 1: 2 stop bits, except for 5-bit character then 1-1/2 bits	RW
1:0	WLS	**Word Length Select.** 0b00: 5-bit character 0b01: 6-bit character 0b10: 7-bit character 0b11: 8-bit character	RW

23.7.2.9 UART Modem Control Register (UMCR,0x010)

The UMCR uses the modem control pins RTS_ and CTS_ to control the interface with a modem or data set. UMCR also controls the loopback mode. Loopback mode must be enabled before the UART is enabled..

	UMCR																BASE + 0x010													
Bit	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
	Reserved																								MDCE	FCM	Reserved	LOOP	Reserved		RTS	
RST	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

Bits	Name	Description	RW
7	MDCE	**Modem Control Enable.** 0: Modem function is disabled 1: Modem function is enabled	RW
6	FCM	**Flow Control Mode.** 0: Flow control by software 1: Flow control by hardware	RW
5	Reserved	Writing has no effect, read as zero.	R
4	LOOP	**Loop Back.** This bit is used for diagnostic testing of the UART. When LOOP is 1, TXD output pin is set to a logic 1 state, RXD is disconnected from the pin, and the output of the transmitter shifter register is looped back into the receiver shift register input internally, similar to CTS_ and RTS_ pins and the RTS bit of the UMCR is connected to CTS bit of UMSR respectively. Loopback mode must be selected before the UART is enabled. 0: Normal operation mode	RW

		1: Loopback-mode UART operation	
3:2	Reserved	Writing has no effect, read as zero.	R
1	RTS	**Request To Send.** This bit can control the RTS_ output state. 0: RTS_ force to high 1: RTS_ force to low	RW
0	Reserved	Writing has no effect, read as zero.	R

23.7.2.10 UART Line Status Register (ULSR,0x014)

The read-only ULSR indicates status information during the data transfer. Receive error information in ULSR[4:1] remains set until software reads ULSR and it must be read before the error character is read.

	ULSR																													BASE + 0x014
Bit	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
	Reserved																								FIFOE	TEMP	TDRQ	BI	FMER	PARER	OVER	DRY
RST	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	0	0	0	0	0

Bits	Name	Description	RW
7	FIFOE	**FIFO Error Status. (FIFO mode only)** FIFOE is set when there is at least one kind of receive error (parity, frame, overrun, break) for any of the characters in receive buffer. FIFOE is reset when all error characters are read out of the buffer. During DMA transfer, the error interrupt generates when FIFOE is 1, and no receive DMA request generates even when data in receive buffer reaches the trigger threshold until all the error characters are read out. In non-DMA mode, FIFOE set does not generate error interrupt. 0: No error data in receive buffer or non-FIFO mode 1: One or more error character in receive buffer	R
6	TEMP	**Transmit Holding Register Empty.** Set when both UTHR and shift register are empty. It is cleared when either the UTHR or the shift register contains a data character. 0: There is data in the transmit shifter and UTHR 1: All the data in the transmit shifter and UTHR has been shifted out	R
5	TDRQ	**Transmit Data Request.** Set when UTHR has half or more empty location (FIFO mode) or empty	R

		(non-FIFO mode). When both UIER.TDRIE and TDRQ are 1, transmit data request interrupt generates or during DMA transfer, DMA request to the DMA controller generates when UIER.TDRIE is 0 and TDRQ is 1. 0: There is one (non-FIFO mode) or more than half data (FIFO mode) in UTHR 1: None data (non-FIFO mode) or half or less than half data (FIFO mode) in UTHR	
4	BI	**Break Interrupt.** BI is set when the received data input is held low for longer than a full-word transmission time (the total time of start bit + data bits + parity bit + stop bits). BI is cleared when the processor reads the ULSR. In FIFO mode, only one character equal to 0x00 is loaded into the FIFO regardless of the length of the break condition. BI shows the break condition for the character at the front of the FIFO, not the most recently received character. 0: No break signal has been received 1: Break signal received	R
3	FMER	**Framing Error.** Set when the bit following the last data bit or parity bit is detected to be 0. If the ULCR had been set for two or one and half stop bits, the other stop bits are not checked except the first one. In FIFO mode, FMER shows a framing error for the character at the front of the receive buffer, not for the most recently received character. Cleared when the processor reads the ULSR. 0: No framing error 1: Invalid stop bit has been detected	R
2	PARER	**Parity Error.** Indicates that the received data character does not have the correct even or odd parity, as selected by the even parity select bit. PARER is set upon detection if a parity error and is cleared when the processor reads the ULSR. In FIFO mode, PARER shows a parity error for the character at the front of the FIFO, not the most recently received character. 0: No parity error 1: Parity error has occurred	R
1	OVER	**Overrun Error.** Set when both receive buffer and shifter are full and new data is received which will be lost. Cleared when the processor reads the ULSR. 0: No data has been lost 1: Receive data has been lost	R

0	DRY	**Data Ready.** Set when a complete incoming character has been received into the Receive Buffer registers. DRY is cleared when the receive buffer is read (non-FIFO mode) or when the buffer is empty or when the buffer is reset by setting UFCR.RFRT to 1. 0: No data has been received 1: Data is available in URBR	R

23.7.2.11 UART Modem Status Register (UMSR,0x018)

The read-only UMSR provides the current state of the control lines from the modem to the processor. They are cleared when the processor reads UMSR.

	UMSR																BASE + 0x018													
Bit	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
	Reserved																								Reserved			CTS	Reserved			CCTS
RST	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	0	0	0	0	0

Bits	Name	Description	RW
7:5	Reserved	Writing has no effect, read as zero.	R
4	CTS	**Status of Clear To Send.** When MDCE bit is 1, this bit is the complement of CTS_ input. If Loop bit of UMCR is 1, this bit is equivalent to RTS bit of UMCR. 0: CTS_ pin is 1 1: CTS_ pin is 0	R
3:1	Reserved	Writing has no effect, read as zero.	R
0	CCTS	**Change status of CTS_.** When MDCE bit is 1, this bit indicates the state change on CTS_ pin. 0: No state change on CTS_ pin since last read of UMSR 1: A change occurs on the state of CTS_ pin	R

23.7.2.12 UART Scratchpad Register(USPR,0x01C)

This Scratchpad register is used as a scratch register for the programmer and has no effect on the UART.

	USPR																BASE + 0x01C													
Bit	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

	Reserved	Scratch Data
RST	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	

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect, read as zero.	R
7:0	STCH_D	Be scratched data	RW

23.7.2.13 Infrared Selection Register (ISR,0x020)

The ISR is used to configure the slow-infrared (SIR) interface that is provided in each UART to support two-way wireless communication using infrared transmission that conforms to the IrDA serial infrared specification 1.1. The maximum frequency is up to 115.2kbps.

	ISR																BASE + 0x020													
Bit	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
	Reserved																								Reserved			RDPL	TDPL	XMODE	RCVEIR	XMITIR
RST	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

Bits	Name	Description	RW
31:5	Reserved	Writing has no effect, read as zero.	R
4	RDPL	**Receive Data Polarity.** 0: Slow-infrared (SIR) interface decoder takes positive pulses as zeros 1: SIR decoder takes negative pulses as zeros	RW
3	TDPL	**Transmit Data Polarity.** 0: SIR encoder generates a positive pulse for a data bit of zero 1: SIR encoder generates a negative pulse for a data bit of zero	RW
2	XMODE	**Transmit Pulse Width Mode.** Set when the transmit encoder needs to generate 1.6us pulses (that are 3/16 of a bit-time at 115.2 kbps). Cleared when the transmit encoder needs to generate 3/16 of a bit-time wide according to current baud rate. 0: Transmit pulse width is 3/16 of a bit-time wide 1: Transmit pulse width is 1.6 us	RW
1	RCVEIR	**Receiver SIR Enable.** This bit is used to select the signal from the RXD pin is processed by the IrDA decoder before it is fed to the UART (RCVEIR = 1) or bypass IrDA decoder and is fed directly to the UART (RCVEIR = 0).	RW

		0: Receiver is in UART mode 1: Receiver is in SIR mode	
0	XMITIR	**Transmitter SIR Enable.** This bit is used to select TXD output pin is processed by the IrDA encoder before it is fed to the device pin (XMITIR = 1) or bypass IrDA encoder and is fed directly to the device pin (XMITIR = 0). **NOTE:** disable infrared LED before XMITIR is set, otherwise a false start bit may occur. 0: Transmitter is in UART mode 1: Transmitter is in SIR mode	RW

23.7.2.14 UART M Register (UMR,0x24)

It will take UART at least M cycles to transmit or receive one bit.

It will take UART at most M+1 cycles to transmit or receive one bit.

	UMR																BASE + 0x024													
Bit	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
	Reserved																								Reserved			M				
RST	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

Bits	Name	Description	RW
31:5	Reserved	Writing has no effect, read as zero.	R
4:0	M	The value of UMR register.	W

23.7.2.15 UART Add Cycle Register (UACR,0x028)

It controls time that UART need to transmit or receive the nth bit of the data;

	UACR																BASE + 0x028													
Bit	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
	Reserved																				AC											
RST	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

Bits	Name	Description	RW
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	AC	If the nth bit of the register is：	R

		1 ： it will take UART M+1 cycles to transmit or receive the bit of data; 0 ： it will take UART M cycles to transmit or receive the bit of data For except; 12'h0 : it will take UART M cycles to transmit or receive one bit of data; 12'hFFF : it will take UART M+1 cycles to transmit or receive one bit of data;	

23.7.2.16 UART RXFIFO Counter Register (URCR,0x040)

Indicates there are n data in RXFIFO.

	URCR																													BASE + 0x040
Bit	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
	Reserved																								Reserved	RCNT						
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:7	Reserved	Writing has no effect, read as zero.	R
6:0	RCNT	RXFIFO Counter. Indicates there are n data in RXFIFO when this field is n.	R

23.7.2.17 UART TXFIFO Counter Register (UTCR,0x044)

Indicates there are n data in TXFIFO

	UTCR																													BASE + 0x044
Bit	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
	Reserved																								Reserved	TCNT						
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:7	Reserved	Writing has no effect, read as zero.	R
6:0	TCNT	TXFIFO Counter. Indicates there are n data in TXFIFO when this field is n.	R

23.8 Operation Flow

The following sections describe the UART operations that include flow of configuration, data transmission, data reception, and Infrared mode.

23.8.1 UART Configuration

Before UART starts to transfer data or changing transfer format, configuration must be done to define the transfer format. The sample flow is as the following:

In FIFO mode, set FME bit of UFCR to 1, reset receive and transmit FIFO, then initialize the UART as described below:

1 Clear UFCR.UME to 0.
2 Set value in UDLL/UDHR to generate the baud rate clock.
3 Set data format in ULCR.
4 If it is in FIFO MODE, set FME bit and other FIFO control in UFCR, reset receive and transmit FIFO, otherwise skip item 4.
5 Set each interrupt enable bit in UIER in interrupt-based transfer or set UFCR.DME in DMA-based transfer (DMA transfer is FIFO mode only), then set UFCR.UME.

23.8.2 Data Transmission

After configuration, UART is ready for data transfer. For data transmission, refer to the following procedure:

1 Read ULSR.TDRQ (interrupt disable) or wait for transmit data request interrupt (interrupt enable), if TDRQ = 1 or transmit data request interrupt generates, that means there is enough empty location in UTHR for new data.
2 If ULSR.TDRQ is 1 or get the transmit data request interrupt, write transmit data to UTHR to start transmission.
3 Do item 1 and item 2 if there are more data waiting for transmit.
4 After all necessary data are written to UTHR, wait ULSR.TEMP = 1, that means all data completely transmitted.
5 If it is necessary to send break, set ULCR.SBK and at least wait for 1-bit interval time to send a valid break, then clear ULCR.SBK.
6 Clear UME bit to finish UART transmission.

23.8.3 Data Reception

After configuration, UART is ready for data transfer. For data reception, refer to the following sample procedure:

1 Read ULSR.DRY (interrupt disable) or wait for receive data request interrupt (interrupt enable), if ULSR.DRY =1 or receive data request interrupt generates, that means URBR has one data (non-FIFO mode) or data in URBR reaches the trigger value. (FIFO mode)
2 If ULSR.DRY = 1 or receive data request interrupt generates, then read ULSR.FIFOE or see if there is error interrupt, if FIFOE = 1, it means received data has receive error, then go to error

handler, other wise go to item 3.

3 Read one received data in URBR (non-FIFO mode) or data equal to trigger value in URBR. (FIFO mode)

4 Check whether all data received: check whether ULSR.DRY = 0, in FIFO mode and interrupt is enabled, timeout interrupt may generate, when timeout interrupt generates, read URBR till ULSR.DRY = 0.

5 Clear UFCR.UME to end data reception when all data are received and ULSR.DRY = 0.

23.8.4 Receive Error Handling

A sample error handling flow is as the following:

1 If ULSR.FIFOE = 1, it means there is receive error in received data, then check what error it is.

2 If ULSR.OVER = 1, go to OVER error handling.

3 If ULSR.BI = 1, go to Break handling.

4 If ULSR.FMER = 1, go to Frame error handling.

5 If PARER = 1, go to PARER error handling.

23.8.5 Modem Transfer

When UMCR.MDCE = 1, modem control is enabled. Transfer flow can be stopped and restarted by software through RTS_ and CTS_ pin. When UART transmitter detects low level on CTS_ pin, it stops transmission and TxD pin goes to mark state after finishing transmitting the current character until it detects CTS_ pin goes back to high level. RTS_ pin is output to receiving UART and its state can be controlled by setting UMCR.RTS bit, that is, setting UMCR.RTS to 1, RTS_ pin is low level output that means UART is ready to receive data, on the contrary, it means UART currently can't receive more data.

23.8.6 DMA Transfer

UART can operate in DMA-based (UFCR.DME = 1, FIFO mode only), that is, dma request initiated by UART takes the place of interrupt request and transmission/reception is carried out using DMA instead of CPU. Be sure that software guarantee to disable transmit and receive interrupt except timeout and error interrupts.

During DMA transfer, if an interrupt occurs, software must first read the ULSR to see if an error interrupt exists, then check the UIIR for the source of the interrupt and if DMA channel is already halt because of the error indicator from UART, then disable DMA channel and read out all the error data from receive FIFO. Software re-set and re-enable DMA and data transfer by DMA will re-start.

23.8.7 Slow IrDA Asynchronous Interface

Each UART supports slow infra-red (SIR) transmission and reception by setting ISR.XMITIR and ISR.RCVEIR to 1 (make sure the two bits are not set to 1 at the same time because SIR can't operate

full-duplex). According to the IrDA 1.1, data rate is limited at a maximum value of 115.2Kbps.

In SIR transmit mode, the transmit pulse comes out at a rate of 3/16 (when the transmit data bit is zero); in SIR receive mode, the receiver must detect the 3/16 pulsed period to recognize a zero value (an active high or low pulse is demodulation to 0, and no pulse is demodulation to 1).

Compared to normal UART, there are some limitations to SIR, that is, each character is fixed to 8-bit data width, no parity and 1 stop bit and modem function is ignored. The IrDA 1.1 specifies a minimum 10ms latency after an optical node ceases transmitting before its receiver recovers its receiving function and software must guarantee this delay.

In the IrDA 1.1 specification, communication must start up at the rate of 9600bps, but then allows the link to negotiate higher (or lower) data rates if supported by both ends. However, the communication rate will not automatically change. Change, if necessary, is performed by software.

23.8.8 For any frequency clock to use the UART

NOTE: if you don't set M register and UACR the UART work at normal mode with the specified frequencies. To use other frequency you should to set M register and UACR to right value.

1 The Improving

Following changes are made:

a One bit is composed by M CLK_{BR} cycles, which can be 4~1024.

b Some extra CLK_{BR} cycles can be inserted in some bits in one frame, so that like M has fraction.

For instance:

$CLK_{BR} = CLK_{DEV} / N$ N = 1, 2, …

$CLK_{BR} = CLK_{DEV}$ = 4MHz

Band rate = 460800

In accurate

$M_a = 8.681$

We take

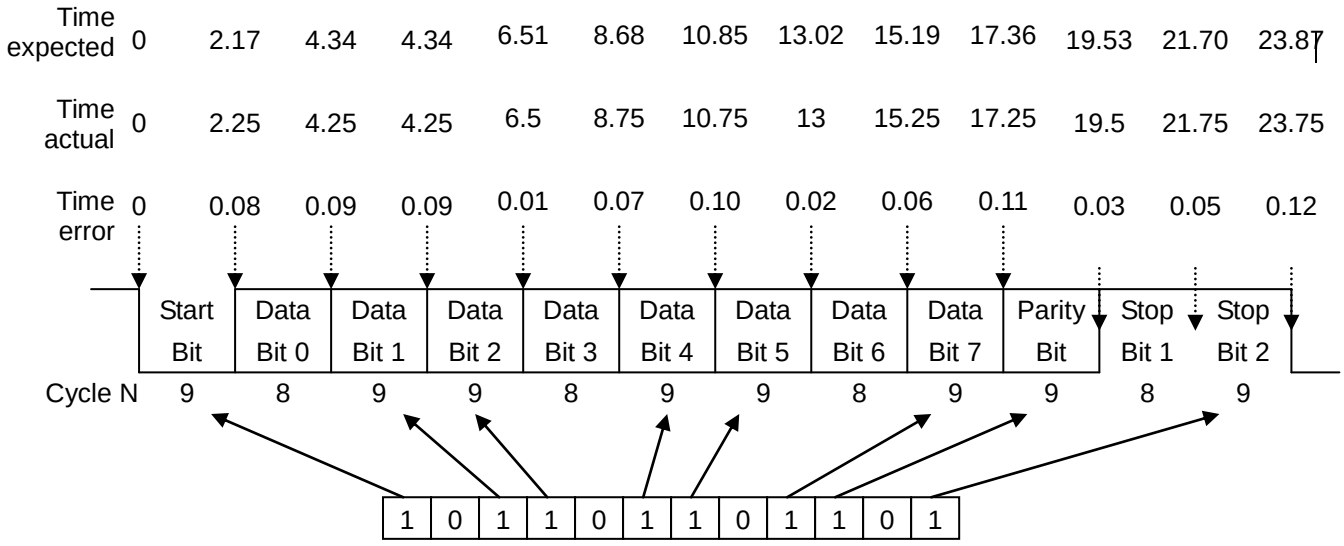
M = 8, with 8 extra cycles in every frame

A 12-bit register is used to indicate where to insert the extra cycles.

The first line is the time expected

The second line is the time actual

The third line is the time error



For transmission, in theory, the biggest error is half of CLK_{BR} cycle, which is 0.125us here.

2 To set UMR register

$CLK_{BR} = CLK_{DEV} / N$

$M_a = CLK_{BR}$/band rate

M is modem of M_a.

Write M to Mregister.

Considering the power and the robust quality, for M form 6 to 32 is you better select by set the UDLR.

The max error

$$\frac{0.5/CLK_{BR}}{M_a/CLK_{BR}} = 0.5/M_a < 0.5/M$$

M	4	8	16	32	64
error/W_{bit}	12.5%	6.25%	3.125%	1.56%	0.78%

3 To set UACR value

For each bit of it means:

0: means not to add additional cycle to the bit that UART is prepare to transmit or receive, in another word, you will to use M cycles to transmit or receive the bit

1: means to add additional cycle to the bit that UART is prepare to transmit or receive, in another word, you will to use M+1 cycles to transmit or receive the bit

To set UACR value you must ensure that the max error of each bit should be less than $0.5P_{BR}$.

For example：M_a-M =0.15; M+1-M_a=0.85;

Write 8 to UMR,

Write 0x408 to UACR

cycle/bit	：	M，	M，	M，	M+1，	M，	M，	M，	M，	M，	M，	M+1，	M
UACR	：	0	0	0	1	0	0	0	0	0	0	1	0

23.9 Index

Table 23-5 Description of Proprietary Vocabulary

Name	Description
UART	universal asynchronous receiver/transmitter
SIR	Serial infrared

24 MMC/SD CE-ATA Controller（MSC）

24.1 Overview

The Multi Media Card (MMC) is a universal low cost mass storage and communication media that is designed to cover a wide area of applications such as electronic toys, organizers, PDAs, smart phones, and so on.

The Secure Digital (SD) card is an evolution of MMC. It is specifically designed to meet the security, capacity, performance, and environmental requirements inherent in newly emerging audio and video consumer electronic devices. The physical form factor, pin assignment, and data transfer protocol are forward compatible with the Multi Media Card with some additions. An SD card can be categorized as SD memory or SD I/O card, commonly known as SDIO. A memory card invokes a copyright protection mechanism that complies with the security of the SDMI standard and is faster and capable of higher memory capacity. The SDIO card provides high-speed data I/O with low-power consumption for mobile electronic devices.

For CE-ATA detail protocol , please referred to WWW.CE-ATA.ORG.

24.2 Features

- Fully compatible with the MMC System Specification version 4.2
- Support SD Specification 3.0
- Support SD I/O Specification 1.0 with 1 command channel and 4 data channels
- Maximum data rate is 50MBps
- Support MMC data width 1bit ,4bit and 8bit
- Built-in programmable frequency divider for MMC/SD bus
- Built-in Special Descriptor DMA
- Multi-SD function support including multiple I/O and combined I/O and memory
- IRQ supported enable card to interrupt MMC/SD controller
- Single or multi block access to the card including erase operation
- Stream access to the MMC card
- Supports SDIO read wait, interrupt detection during 1-bit or 4-bit access
- Supports CE-ATA digital protocol commands
- Support Command Completion Signal and interrupt to CPU
- Command Completion Signal disable feature
- The maximum block length is 4096bytes

24.3 Block Diagram

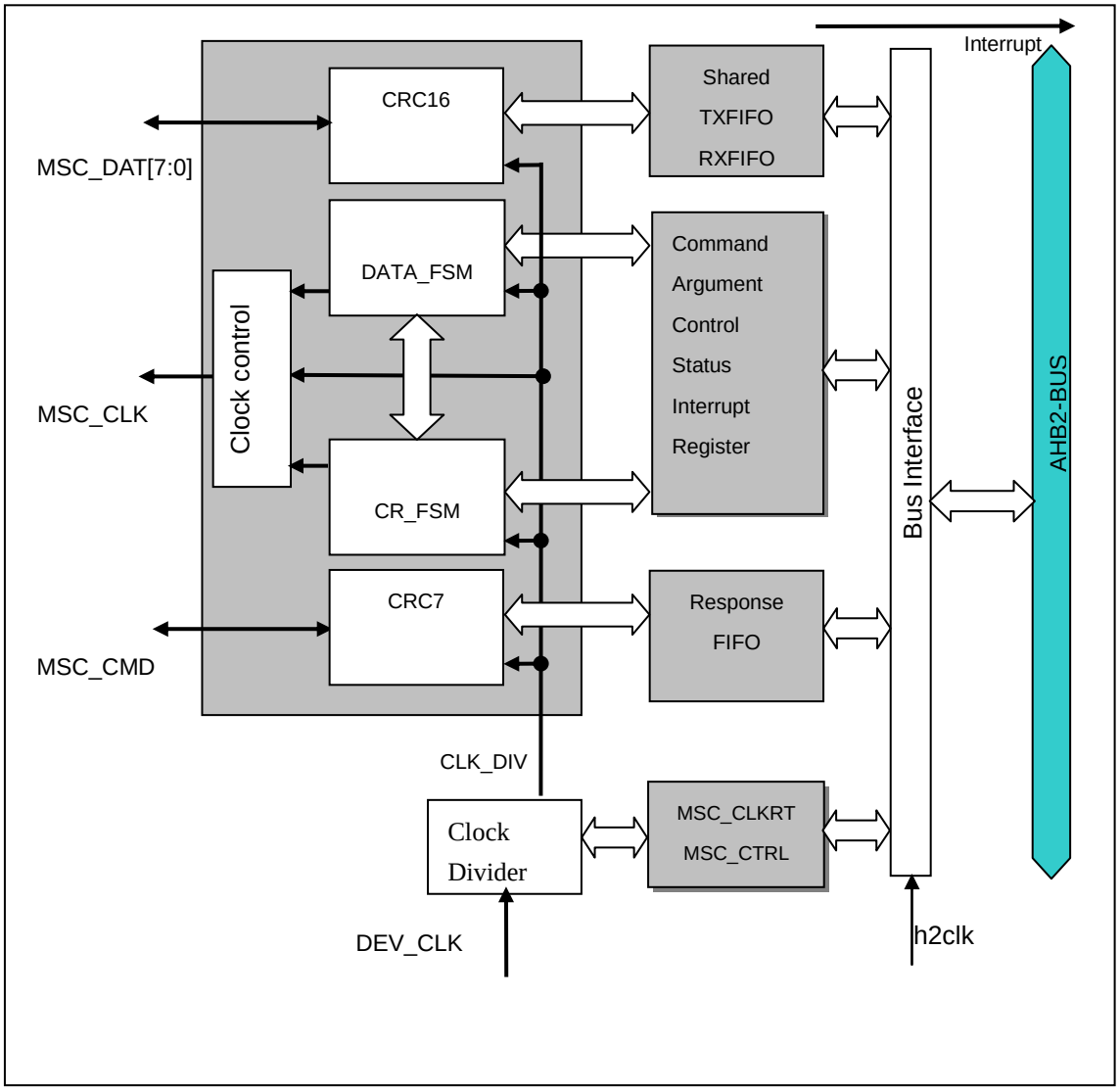


Figure 24-1 MMC/SD Controller Block Diagram

24.4 Functional Description

All communication between system and cards is controlled by the MSC. The MSC sends commands of two type: broadcast and addressed (point-to-point) commands.

Broadcast commands are intended for all cards, command like “Go_Idle_State”, “Send_Op_Cond”, “All_Send_CID” and “Set_Relative_Addr” are using way of broadcasting. During Broadcast mode, all cards are in open-drain mode, to avoid bus contention.

After Broadcast commands “Set_Relative_Addr” issue, cards are enter standby mode, and Addressed command will be used from now on, in this mode, CMD/DAT will return to push-pull mode, to have maximum driving for maximum operation frequency.

The MMC and the SD are similar product. Besides the 4x bandwidth and the built-in encryption, they are being programmed similarly.

The MMC/SD controller (MSC) is the interface between the software and the MMC/SD bus. It is responsible for the timing and protocol between the software and the MMC/SD bus. It consists of control and status registers, a 16-bit response FIFO that is 8 entries deep, and one 32-bit receive/transmit data FIFOs that are 16 entries deep. The registers and FIFOs are accessible by the software.

MSC also enable minimal data latency by buffering data and generating and checking CRCs.

24.4.1 MSC Reset

The MMC/SD controller (MSC) can be reset by a hardware reset or software reset. All registers and FIFO controls are set to their default values after any reset.

24.4.2 Voltage Validation

All cards shall be able to establish communication with the host using any operation voltage in the maximal allowed voltage range specified in this standard. However, the support minimum and maximum values for Vdd are defined in Operation Conditions register (OCR) and many not cover the whole range. Cards that store the CID and CSD data in the payload memory would be able to communicate these information only under data transfer Vdd conditions. That means if host and card have non compatible Vdd ranges, the card will not be able to complete the identification cycle, nor to send CSD data.

Therefore, a special command Send_Op_Cont (CMD1 for MMC), SD_Send_Op_Cont (CMD41 for SD Memory) and IO_Send_Op_Cont (CMD5 for SDIO) are designed to provide a mechanism to identify and reject cards which do not match the Vdd range desired by the host. This is accomplished by the host sending the required Vdd voltage window as the operand of this command. Cards which can not perform data transfer in the specified range must discard themselves from further bus operations and go into Inactive State. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before sending out-of-range cards into the Inactive State. This query should be used if the host is able to select a common voltage range or if a notification to the application of non usable cards in the stack is desired.

24.4.3 Card Registry

Card registry on MCC and SD card are different.
For SD card, Identification process start at clock rate Fod, while CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated the host will request the cards to send their valid operation conditions. The response to ACMD41 is the operation condition register of the card. The same command shall be send to all of the new cards in the system. Incompatible cards are sent into Inactive State. The host then issue the command All_Send_CID (CMD2) to each card and get its unique card identification (CID) number. Card that is unidentified, that is, which is in Ready State, send

its CID number as the response. After the CID was sent by the card it goes into Identification State. Thereafter, the host issues Send_Relative_Addr (CMD3) asks the card to publish a new relative card address (RCA), which is shorter that CID and which will be used to address the card in the future data transfer mode. Once the RCA is received the card state changes to the Stand-by State. At this point, if the host wants that the card will have another RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card. The host repeats the identification process, that is, the cycles with CMD2 and CMD3 for each card in the system.

In MMC, the host starts the card identification process in open-drain mode with the identification clock rate Fod. The open drain driver stages on the CMD line allow parallel card operation during card identification. After the bus is active the host will request the cards to send their valid operation conditions (CMD1). The response to CMD1 is the 'wired or' operation on the condition restrictions of all cards in the system. Incompatible cards are sent into Inactive State. The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards, that is, those which are in Ready State, simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit-stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods stop sending their CID immediately and must wait for the next identification cycle. Since CID is unique for each card, only one card can be successfully send its full CID to the host. This card then goes into Identification State. Thereafter, the host issues Set_Relative_Addr (CMD3) to assign to this card a relative card address (RCA). Once the RCA is received the card state changes to the Stand-by State, and the card does not react to further identification cycles, and its output switches from open-drain to push-pull. The host repeat the process, which is CM2 and CMD3, until the host receive time-out condition to recognize completion of the identification process.

24.4.4 Card Access

24.4.4.1 Block Access, Block Write and Block Read

During block write (CMD24-27) one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. A card supporting block write shall always be able to accept a block of data defined by WRITE_BL_LEN. If the CRC fails, the card shall indicate the failure on the DAT line; the transferred data will be discarded and not written, and all further transmitted blocks (in multiple block write mode) will be ignored.

Programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card will report an error and not change any register contents. Some cards may require long and unpredictable times to write a block of data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the DAT line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) at any time, and the card will respond with its status. The status

bit READY_FOR_DATA indicates whether the card can accept new data or whether the write process is still in progress). The host may deselect the card by issuing CMD7 (to select a different card) which will displace the card into the Disconnect State and release the DAT line without interrupting the write operation. When reselecting the card, it will reactivate busy indication by pulling DAT to low if programming is still in progress and the write buffer is unavailable.

Block read is similar to stream read, except the basic unit of data transfer is a block whose maximizes is defined in the CSD (READ_BL_LEN). If READ_BL_PARTIAL is set, smaller blocks whose starting and ending address are entirely contained within one physical block (as defined by READ_BL_LEN) may also be transmitted. Unlike stream read, a CRC is appended to the end of each block ensuring data transfer integrity. CMD17 (READ_SINGLE_BLOCK) initiates a block read and after completing the transfer, the card returns to the Transfer state. CMD18 (READ_MULTIPLE_BLOCK) starts a transfer of several consecutive blocks. Blocks will be continuously transferred until a stop command is issued. If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed, the card shall detect a block misalignment at the beginning of the first mis-aligned block, set the ADDRESS_ERROR error bit in the status register, abort transmission and wait in the Data State for a stop command.

24.4.4.2 Stream Access, Stream Write and Stream Read (MMC Only)

Stream write (CMD20) starts the data transfer from the host to the card beginning from the starting address until the host issues a stop command. Since the amount of data to be transferred is not determined in advance, CRC can not be used. If the end of the memory range is reached while sending data and no stop command has been sent by the host, all further transferred data is discarded.

There is a stream oriented data transfer controlled by READ_DAT_UNTIL_STOP (CMD11). This command instructs the card to send its payload, starting at a specified address, until the host sends a STOP_TRANSMISSION command (CMD12). The stop command has execution delay due to the serial command transmission. The data transfer stops after the end bit of the stop command. If the end of the memory range is reached while sending data and no stop command has been sent yet by the host, the contents of the further transferred payload is undefined.

24.4.4.3 Erase, Group Erase and Sector Erase (MMC Only)

It is desirable to erase many sectors simultaneously in order to enhance the data throughput. Identification of these sectors is accomplished with the TAG_* commands. Either an arbitrary set of sectors within a single erase group, or an arbitrary selection of erase groups may be erase at one time, but not both together. That is, the unit of measure for determining an erase is either a sector or an erase group. If a set of sectors must be erased, all selected sectors must lie within the same erase group. To facilitate selection, a first command with the starting address is followed by a second command with the final address, and all sectors (or groups) within this range will be selected for erase.

24.4.4.4 Wide Bus Selection/Deselection

Wide Bus (4 bit bus width) operation mode may be selected / deselected using ACMD6. The default bus width after power up or GO_IDLE (CMD0) is 1 bit bus width. ACMD6 command is valid in 'trans state' only. That means the bus width may be changed only after a card was selected (CMD7).

24.4.5 Protection Management

Three write protect methods are supported in the host for Cards, Card internal write protect (Card's responsibility), Mechanical write protect switch (Host responsibility only) and Password protection card lock operation.

24.4.5.1 Card Internal Write Protection

Card data may be protected against either erase or write. The entire card may be permanently write protected by the manufacturer or content provider by setting the permanent or temporary write protect bits in the CSD. For cards which support write protection of groups of sectors by setting the WP_GRP_SIZE sectors as specified in the CSD), and the write protection may be changed by the application. The SET_WRITE_PROT command sets the write protection of the addressed write-protect group, and the CLR_WRITE_PROT command clears the write protection of the addressed write-protect group.

The SEND_WRITE_PROT command is similar to a single block read command. The card shall send a data block containing 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits. The address field in the write protect commands is a group address in byte units. The card will ignore all LSB's below the group size.

24.4.5.2 Mechanical write protect switch

A mechanical sliding tablet on the side of the card will be used by the user to indicate that a given card is write protected or not. If the sliding tablet is positioned in such a way that the window is open that means the card is write protected. If the window is close the card is not write protected.

A proper, matched, switch on the socket side will indicated to the host that the card is write protected or not. It is the responsibility of the host to protect the card. The position of the write protect switch is un-known to the internal circuitry of the card.

24.4.5.3 Password Protect

The password protection feature enables the host to lock a card while providing a password, which later will be used for unlocking the card. The password and its size is kept in an 128-bit PWD and 8-bit PWD_LEN registers, respectively. These registers are non-volatile so that a power cycle will not erase them.

Locked cards respond to (and execute) all commands in the basic command class (class 0) and “lock card” command class. Thus the host is allowed to reset, initialize, select, query for status, etc., but not to access data on the card. If the password was previously set (the value of PWD_LEN is not 0) will be locked automatically after power on. Similar to the existing CSD and CID register write commands the lock/unlock command is available in “trans-state” only. This means that it does not include an address argument and the card must be selected before using it. The card lock/unlock command has the structure and bus transaction type of a regular single block write command. The transferred data block includes all the required information of the command (password setting mode, PWD itself, card lock/unlock etc.). Table 24-1 describes the structure of the command data block.

Table 24-1 Command Data Block Structure

Byte #	Bit 7-4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved	ERASE	LOCK_UNLOCK	CLR_PWD	SET_PWD
1	PWDS_LEN				
2	Password Data				
…					
PWDS_LEN + 1					

ERASE – 1 Defines Forced Erase Operation (all other bits shall be 0) and only the command byte is sent.

LOCK/UNLOCK – 1=Locks the card. 0=Unlock the card (note that it is valid to set this bit together with SET_PWD but it is not allowed to set it together with CLR_PWD).

CLR_PWD – 1=Clears PWD.

SET_PWD – 1=Set new password to PWD.

PWD_LEN – Defines the following password length (in bytes).

PWD – The password (new or currently used depending on the command).

The data block size shall be defined by the host before it send the card lock/unlock command. This will allow different password sizes.

The following paragraphs define the various lock/unlock command sequences:

Lock command sequences:

1 Setting the Password.

a Select a card (CMD7), if not previously selected already.

b Define the block length (CMD16), given by the 8bit card lock/unlock mode, the 8 bits password size (in bytes), and the number of bytes of the new password. In case that a password replacement is done, then the block size shall consider that both passwords, the old and the new one, are sent with the command.

c Send Card Lock/Unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode (SET_PWD), the length (PWD_LEN) and the password itself. In case that a password replacement is done, then

the length value (PWD_LEN) shall include both passwords, the old and the new one, and the PWD field shall include the old password (currently used) followed by the new password.

d In case that the sent old password is not correct (not equal in size and content) then LOCK_UNLOCK_FAILED error bit will be set in the status register and the old password does not change. In case that PWD matches the sent old password then the given new password and its size will be saved in the PWD and PWD_LEN fields, respectively.

NOTE:

the password length register (PWD_LEN) indicates if a password is currently set. When it equals 0 there is no password set. If the value of PWD_LEN is not equal to zero the card will lock itself after power up. It is possible to lock the card immediately in the current power session by setting the LOCK/UNLOCK bit (while setting the password) or sending additional command for card lock.

2 Reset the password.

a Select a card (CMD7), if not previously selected already.

b Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.

c Send the card lock/unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode CLR_PWD, the length (PWD_LEN) and the password (PWD) itself (LOCK/UNLOCK bit is don't care). If the PWD and PWD_LEN is set to 0. If the password is not correct then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

3 Locking a card.

a Select a card (CMD7), if not previously selected already.

b Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of currently used password.

c Send the card lock/unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode LOCK, the length (PWD_LEN) and the password (PWD) itself.

If the PWD content equals to the sent password then the card will be locked and the card-locked status bit will be set in the status register. If the password is not correct then LOCK_UNLOCK_FAILED error bit will be set in the status register.

NOTE:

it is possible to set the password and to lock the card in the same sequence. In such case the host shall perform all the required steps for setting the password (as described above) including the bit LOCK set while the new password command is sent. If the password was previously set (PWD_LEN is not 0), then the card will be locked automatically after power on reset. An attempt to lock a locked card or to lock a card that

does not have a password will fail and the LOCK_UNLOCK_FAILED error bit will be set in the status register.

Unlock command sequences:

1 Unlocking the card.

a Select a card (CMD7), if not previously selected already.

b Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.

c Send the card lock/unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode UNLOCK, the length (PWD_LEN) and the password (PWD) itself.

If the PWD content equals to the sent password then the card will be unlocked and the card-locked status bit will be cleared in the status register. If the password is not correct then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

NOTE:

the unlocking is done only for the current power session. As long as the PWD is not cleared the card will be locked automatically on the next power up. The only way to unlock the card is by clearing the password. An attempt to unlock an unlocked card will fail and LOCK_UNLOCK_FAILED error bit will be set in the status register.

2 Forcing Erase.

In case that the user forgot the password (the PWD content) it is possible to erase all the card data content along with the PWD content. This operation is called Forced Erase.

a Select a card (CMD7), if not previously selected already.

b Define the block length (CMD16) to 1 byte (8bit card lock/unlock command). Send the card lock/unlock command with the appropriate data block of one byte on the data line including 16-bit CRC. The data block shall indicate the mode ERASE (the ERASE bit shall be the only bit set).

If the ERASE bit is not the only bit in the data field then the LCOK_UNLOCK_FAILED error bit will be set in the status register and the erase request is rejected. If the command was accepted then ALL THE CARD CONTENT WILL BE ERASED including the PWD and PWD_LEN register content and the locked card will get unlocked.

An attempt to force erase on an unlocked card will fail and LOCK_UNLOCK_FAILED error bit will be set in the status register.

24.4.6 Card Status

The response format R1 contains a 32-bit field named card status. This field is intended to transmit the card's status information (which may be stored in a local status register) to the host. If not specified otherwise, the status entries are always related to the previous issued command.

Table below defines the different entries of the status. The type and clear condition fields in the table are abbreviate as follows:

Type:

E: Error bit.

S: Status bit..

R: Detected and set for the actual command response.

X: Detected and set during command execution. The host must poll the card by issuing the status command in order to read these bits.

Clear Condition:

A: According to the card current state.

B: Always related to the previous command. Reception of a valid command will clear it (with a delay of one command).

C: Clear by read.

Table 24-2 Card Status Description

Bits	Identifier	Type	Description	Clear Condition
31	OUT_OF_RAGE	E R	The command's argument was out of the allowed range for this card. 0: No Error 1: Error	C
30	ADDRESS_ERROR	E R X	A misaligned address which did not match the block length was used in the command. 0: No Error 1: Error	C
29	BLOCK_LEN_ERROR	E R	The transferred block length is not allowed for this, or the number of transferred bytes does not match the block length. 0: No Error 1: Error	C
28	ERASE_SEQ_ERROR	E R	An error in the sequence of erase commands occurred. 0: No Error 1: Error	C
27	ERASE_PARAM	E X	An invalid selection of sectors or groups for erase occurred. 0: No Error 1: Error	C
26	WP_VIOLATION	E R X	Attempt to program a write	C

			protected block. 0: No Protected 1: Protected	
25	CARD_IS_LOCKED	S X	When set, signals that the card is locked by the host. 0: Card unlocked 1: Card locked	A
24	LOCK_UNLOCK_FAILED	E R X	Set when a sequence or password error has been detected in lock/unlock card command or if there was an attempt to access a locked card. 0: No Error 1: Error	C
23	COM_CRC_ERROR	E R	The CRC check of the previous command failed. 0: No Error 1: Error	B
22	ILLEGAL_COMMAND	E R	Command not legal for the card state. 0: No Error 1: Error	B
21	CARD_ECC_FAILED	E X	Card internal ECC was applied but failed to correct the data. 0: normal 1: failure	C
20	CC_ERROR	E R X	Internal card controller error. 0: No Error 1: Error	C
19	ERROR	E R X	A general or an unknown error occurred during the operation. 0: No Error 1: Error	C
18	UNDERRUN	E X	The card could not sustain data transfer in stream read mode. 0: No Error 1: Error	C
17	OVERRUN	E X	The card could not sustain data programming in stream write mode. 0: No Error 1: Error	C
16	CID/CSD_OVERWRITE	E R X	Can be either one of the following errors.	C

			0: No Error 1: Error	
15	WP_ERASE_SKIP	S X	Only partial address space was erased due to existing write protected blocks. 0: No Protected 1 : Protected	C
14	CARD_ECC_DISABLED	S X	The command has been executed without using the internal ECC. 0: enabled 1: disabled	A
13	ERASE_RESET	S R	An erase sequence was cleared before executing because an out of erase sequence command was received. 0: normal 1: set	C
12:9	CURRENT_STATE	S X	The state of the card when receiving the command. If the command execution causes a state change, it will be visible to the host in the response to the next command. The four bits are interpreted as binary coded number between 0 and 15. 0: idle 1: ready 2: identify 3: standby 4: transition 5: data 6: receive 7: program 8 : disable (9 – 15) : received	B
8	READY_FOR_DATA	S X	Corresponds to buffer empty signaling on the bus. 0: No Ready 1: Ready	A
7:6	Reserved	-	-	-
5	APP_CMD	S R	The card will expect ACMD, or indication that the command has been interpreted as ACMD. 0: Disable	C

			1: Enable	
4:0	Reserved	-	-	-

24.4.7 SD Status

The SD status contains status bits that are related to the SD card proprietary features and may be used for future application specific usage. The size of the SD status is one data block of 512bit. The content of this register is transmitted to the Host over the DAT bus along with 16-bit CRC. The SD status is sent to the host over the DAT bus if ACMD13 is sent (CMD55 followed with CMD13). ACMD13 can be sent to a card only in tran_state (card is selected). SD status structure is described in below.

The same abbreviation for *type* and *clear condition* were used as for the Card Status above.

Table 24-3 SD Status Structure

Bits	Identifier	Type	Description	Clear Condition
511:510	DAT_BUS_WIDTH	S R	Shows the currently defined data bus width that was defined by SET_BUS_WIDTH command. 00: 1 (default) 01: Reserved 10: 4 bit width 11: Reserved	A
509	SECURED_MODE	S R	Card is in Secured Mode of operation. 0: Not in the Mode 10: In the mode	A
508:496	Reserved.			
495:480	SD_CARD_TYPE	S R	All 0, is SD Memory cards.	A
479:448	SIZE_OF_PROTECTED_AREA	S R	Size of protected area.	A
447:312	Reserved.			
311:0	Reserved for manufacturer.			

24.4.8 SDIO

I/O access differs from memory in that the registers can be written and read individually and directly without a FAT file structure or the concept of blocks (although block access is supported). These registers allow access to the IO data, control of the IO function, and report on status or transfer I/O data to and from the host.

Each SDIO card may have from 1 to 7 functions plus one memory function built into it. A function is a self contained I/O device. I/O functions may be identical or completely different from each other. All I/O functions are organized as a collection of registers, and there is a maximum of 131,072 registers

possible for each I/O function.

24.4.8.1 SDIO Interrupts

In order to allow the SDIO card to interrupt the host, and interrupt function is added to a pin on the SD interface. Pin number 8 which is used as DAT[1] when operating in the 4 bit SD mode is used to signal the card's interrupt to the host. The use of interrupt is optional for each card or function within a card. The SDIO interrupt is "level sensitive", that is, the interrupt line must be held active (low) until it is either recognized and acted upon by the host or de-asserted due to the end of the Interrupt Period. Once the host has serviced the interrupt, it is cleared via an IO write to the appropriate bit in the CCCR. The interrupt output of all SDIO cards is active low.

As Pin 8 of the card is shared between the IRQ and DAT[1] use in the 4 bit SD mode, and interrupt shall only be sent by the card and recognized by the host during a specific time. The time that a low on Pin 8 will be recognized as an interrupt is defined as the Interrupt Period.

The host here will only sample the level of Pin 8 (DAT[1]/IRQ) into the interrupt detector during the Interrupt Period. At all other times, the host will ignore the level on Pin 8. Note that the Interrupt Period is applicable for both memory and IO operations. The definition of the Interrupt Period is different for operations with single block and multiple block data transfer.

24.4.8.2 SDIO Suspend/Resume

Within a multi-function SDIO or a Combo (Mix IO and Memory) card, there are multiple devices (I/O and memory) that must share access to the SD bus. In order to allow the sharing of access to the host among multiple devices, SDIO and combo cards can implement the optional concept of suspend/resume. In a card supports suspend/resume, the host may temporarily halt a data transfer operation to one function or memory (suspend) in order to free the bus for a higher priority transfer to a different function of memory. Once this higher-priority transfer is complete, the original transfer is re-started where it left off (resume). The host controller here is supported by all IO functions except zero, and the memory of a combo card, and can suspend multiple transactions and resume them in any order desired. IO function zero does not support suspend/resume.

The procedure used to perform the Suspend/Resume operation on the SD bus is:

- The host determines which function currently used the DAT[] line(s).
- The host requests the lower priority or slower transaction to suspend.
- The host checks for the transaction suspension to complete.
- The host begins the higher priority transaction.
- The host waits for the completion of the higher priority transaction.
- The host restores the suspended transaction.

24.4.8.3 SDIO Read Wait

The optional Read Wait (RW) operation is defined only for the SD 1-bit and 4-bit modes. The read wait operation allows a host to signal a card that it is doing a read multiple (CMD53) operation to temporarily stall the data transfer while allowing the host to send commands to any function within the SDIO device. To determine if a card supports the Read Wait protocol, the host must test capability bits in CCCR. The timing for Read Wait is base on the Interrupt Period.

24.4.9 Clock Control

The software should guarantee that the card identification process starts in open-drain mode with the clock rate fod (0 ~ 400khz). In addition, the software should also make the card into interrupt mode with fod (only for MMC). The commands that require fod are CMD0, CMD1, CMD2, CMD3, CMD5, CMD40 and ACMD41. In data transfer mode, the MSC controller can operate card with clock rate fpp (0 ~ 25Mhz).

24.4.10 Application Specified Command Handling

The MMC/SD system is designed to provide a standard interface for a variety applications types. In this environment it is anticipate that there will be a need for specific customers/applications features. To enable a common way of implementing these features, two types of generic commands are defined in the standard: Application Specific Command, ACMD, and General Command, GEN_CMD.
GEN_CMD, this command, when received by the card, will cause the card to interpret the following command as an application specific command, ACMD. The ACMD has the same structure as of regular MMC standard commands and it may have the same CMD number. The card will recognize it as ACMD by the fact that it appears after APP_CMD.
The only effect of the APP_CMD is that if the command index of the, immediately, following command has an ACMD overloading, the none standard version will used. If, as an example, a card has a definition for ACMD13 but not for ACMD7 then, if received immediately after APP_CMD command, Command 13 will be interpreted as the non standard ACMD13 but, command 7 as the standard CMD7.
In order to use one of the manufacturer specific ACMD's the host will:

1 Send APP_CMD. The response will have the APP_CMD bit (new status bit) set signaling to the host that ACMD is now expected.
2 Send the required ACMD. The response will have the APP_CMD bit set, indicating that the accepted command was interpreted as ACMD. If a non-ACMD is sent then it will be respected by the card as normal MMC command and the APP_CMD bit in the Card Status stays clear.

If a non valid command is sent (neither ACMD nor CMD) then it will be handled as a standard MMC illegal command error.
The bus transaction of the GEN_CMD is the same as the single block read or write commands (CMD24 or CMD17). The difference is that the argument denotes the direction of the data transfer (rather than the address) and the data block is not a memory payload data but has a vendor specific format and meaning.
The card shall be selected before sending CMD56. The data block size is the BLOCK_LEN that was defined with CMD16. The response to CMD56 will be R1b (card status + busy indication).

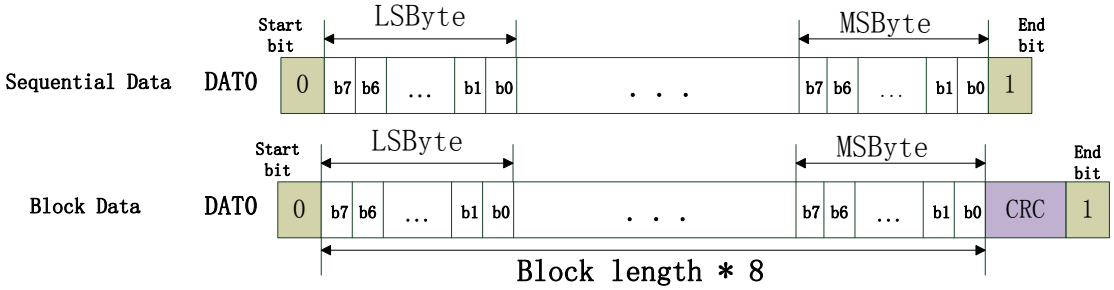
24.5 Pins Description

- MSC_CLK, output, host to card clock signal.
- MSC_CMD, inout, bidirectional command/response signal.
- MSC_DAT[7:0], inout, bidirectional data bus.

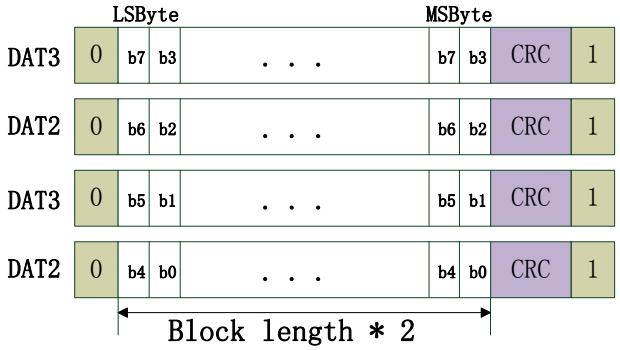
24.6 Data Format Description

MSC0 supports three kinds of data bits width:1bits, 4bits, 8bits. MSC1 and MSC2 only support 1bits and 4bits.

1 bit bus(only DAT0 used):



4 bits bus(DAT3 to DAT0 used):



8 bits bus(DAT7 to DAT0 used):

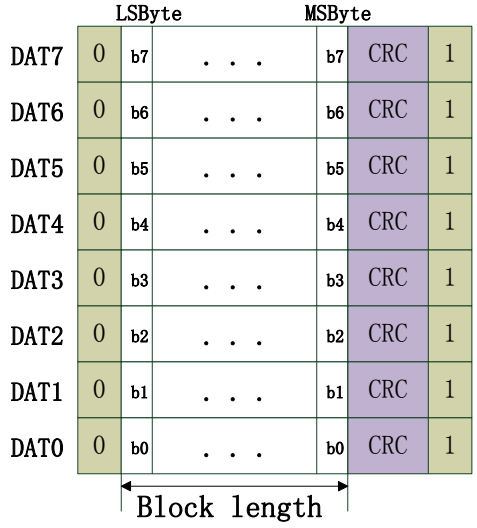


Figure 24-2 msc data format

24.7 Register Description

Following chapter will describe the functions of all software accessible registers.

Conventions:

1. Register Address = Base + Address offset
2. Register read/write attribute
   R- Read only
   W- Write only
   RW- Read and Write
   RCW- Read and Write, but Clear to 0 by Read
   RSW- Read and Write, but Set to 1 by Read
   RWC- Read and Write, Clear to 0 by Write 1, Write 0 has no effect
   RWS- Read and Write, Set to 1 by Write 1, Write 0 has no effect
   RC- Read only, and Clear to 0 by read
   RS- Read only, and Set to 1 by read
3. Reset Value
   1 - reset to 1
   0 - reset to 0
   ? - value unknown after reset

24.7.1 Register Memory Map

Table 24-4 Registers Memory Map Base Address

Name	Base	Description
MSC0	0x13450000	Address base of MSC0
MSC1	0x13460000	Address base of MSC1
MSC2	0x13470000	Address base of MSC2

The MMC-SD-CE_ATA controller is controlled by a set of registers that the application configures before every operation. Table 24-5 lists all the MSC registers.

Table 24-5 MSC Registers Map

Name	RW	Reset Value	Address Offset	Access Size
MSC_CTRL	W	0x0000	0x00	16
MSC_STAT	R	0x??000040	0x04	32
MSC_CLKRT	RW	0x0000	0x08	16
MSC_CMDAT	RW	0x00005000	0x0C	32
MSC_RESTO	RW	0x0100	0x10	16
MSC_RDTO	RW	0x00FFFFFF	0x14	32
MSC_BLKLEN	RW	0x0000	0x18	16
MSC_NOB	RW	0x0000	0x1C	16
MSC_SNOB	R	0x????	0x20	16
MSC_IMASK	RW	0xFFFFFFFF	0x24	32
MSC_IFLG	RW	0x2000	0x28	32
MSC_CMD	RW	0x00	0x2C	8
MSC_ARG	RW	0x00000000	0x30	32

MSC_RES	R	0x????	0x34	16
MSC_RXFIFO	R	0x????????	0x38	32
MSC_TXFIFO	W	0x????????	0x3C	32
MSC_LPM	RW	0x00000000	0x40	32
MSC_DMAC	RW	0x00000000	0x44	32
MSC_DMANDA	RW	0x00000000	0x48	32
MSC_DMADA	R	0x00000000	0x4C	32
MSC_DMALEN	R	0x00000000	0x50	32
MSC_DMACMD	R	0x00000000	0x54	32
MSC_CTRL2	RW	0x00800000	0x58	32
MSC_RTCNT	R	0x00000000	0x5C	32

24.7.2 Register and Fields Description

24.7.2.1 MSC Control Register (MSC_CTRL,0x00)

	MSC_CTRL																BASE + 0x00													
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	SEND_CCSD	SEND_AD_CCSD	Reserved						EXIT_MULTIPLE	EXIT_TRANSFER	START_READ WAIT	STOP_READ WAIT	RESET	START_OP	CLOCK_CTRL	
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15	SEND_CCSD	0: clear bit 1: Send Command Completion Signal Disable (CCSD) to CE_ATA device when set, host sends CCSD to CE_ATA device. Software set the bit only if current command is expecting CCS and interrupts are enabled in CE_ATA devices. Once the CCSD pattern is sent to device, host automatically clears the SEND_CCSD bit.	W
14	SEND_AS_CCSD	0: clear bit 1: send internally generated stop after sending CCSD to CE_ATA device When set, host automatically sends internally-generated STOP command(CMD12) to CE_ATA device. After sending CMD12, MSC_STAT. AUTO_CMD12_DONE is set and generates interrupt to CPU. After sending the CCSD, controller automatically clears the SEND_AS_CCSD bit.	W

13:8	Reserved	Writing has no effect.	R
7	EXIT_MULTIPLE	If CMD12 or CMD52 (I/O abort) is to be sent to terminate multiple block read/write in advance, set this bit to 1. 0: No effect 1: Exit from multiple block read/write	W
6	EXIT_TRANSFER	**Only used for SDIO suspend/resume and MMC stream read.** For SDIO, after suspend is accepted, set this bit with 1. For MMC, after the data of the expected number are received, set this bit with 1. 0: No effect 1: Exit from multiple block read/write after suspend is accepted, or exit from stream read	W
5	START_READ WAIT	Only used for SDIO Read Wait. Start the Read Wait cycle. 0: No effect 1: Start Read Wait	W
4	STOP_READ WAIT	Only used for SDIO Read Wait. Stop the Read Wait cycle. 0: No effect 1: Start Read Wait	W
3	RESET	Reset the MSC controller. 0: No effect 1: Reset the MSC controller	W
2	START_OP	This bit is used to start the new operation. When starting the clock, this bit can be 1. When stopping the clock, this bit can only be 0. 0: Do nothing 1: Start the new operation	W
1:0	CLOCK_CTRL	These bits are used to start or stop clock. 00: Do nothing 01: Stop MMC/SD clock 10: Start MMC/SD clock 11: Reserved	W

24.7.2.2 MSC Status Register (MSC_STAT,0x04)

	MSC_STAT																BASE + 0x04														
Bit	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
	AUTO_CMD12_DONE	Reserved		PIN_LEVEL					Reserved			BCE	BDE	BAE	BAR	DMAEND	IS_RESETTING	SDIO_INT_ACTIVE	PRG_DONE	DATA_TRAN_DONE	END_CMD_RES	DATA_FIFO_AFULL	IS_READ WAIT	CLK_EN	DATA_FIFO_FULL	DATA_FIFO_EMPTY	CRC_RES_ERR	CRC_READ_ERROR	CRC_WRITE_ERRO	R		TIME_OUT_RES	TIME_OUTREAD
RST	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

Bits	Name	Description	RW
31	AUTO_CMD12_DONE	Indicates that the stop command (CMD12) that is internally generated by controller has finished.	R
30:29	Reserved	Read as zero.	R
28:24	PIN_LEVEL	MSC Interface Pin Level PIN_LEV[4:0] indicates the pin level of MSC_CMD, MSC_DAT[3:0] separately.	R
23:21	Reserved	Read as zero.	R
20	BCE	Boot CRC error. 0: No boot CRC error occurs. 1: Boot CRC is not correct.	R
19	BDE	Boot data end. 0: No boot data or boot data is not finished. 1: Boot data is received completely.	R
18	BAE	Boot acknowledge is error. 0: No boot acknowledge is received or boot acknowledge is correct (decided by BAR) 1: The received boot acknowledge is not correct.	R
17	BAR	Boot acknowledge received. 0: No boot acknowledge received. 1: Boot acknowledge received.	R
16	DMAEND	Indicates that the DMA has finished the current transfer.	R
15	IS_RESETTING	MSC is resetting after power up or MSC_CTRL[RESET] is written with 1. 0: Reset has been finished 1: Reset has not been finished	R
14	SDIO_INT_ACTIVE	Indicates whether an interrupt is detected at the SD I/O card. A separate acknowledge command to the card is required to clear this interrupt. 0: No interrupt detected 1: The interrupt from SDIO is detected	R
13	PRG_DONE	Indicates whether card has finished programming. 0: not finished 1: finished	R
12	DATA_TRAN_DONE	Indicates whether data transmission to card has completed. 0: not completed 1: completed	R
11	END_CMD_RES	Indicates whether command and response/no-response sequence have been completed 0: not completed 1: completed	R

10	DATA_FIFO_AFULL	Indicates whether the data FIFO is almost full. 0: The number of words in FIFO is less then 127. 1: The number of words in FIFO is equal to or great then 127	R
9	IS_READ WAIT	Indicates whether SDIO card has entered Read Wait State. 0: Card has not entered Read Wait 1: Card has entered Read Wait	R
8	CLK_EN	Clock enabled. 0: Clock is off 1: Clock is on	R
7	DATA_FIFO_FULL	Indicates whether the data FIFO is full. 0: Data FIFO is not full 1: Data FIFO is full	R
6	DATA_FIFO_EMPTY	Indicates whether data FIFO is empty. 0: Data FIFO is not empty 1: Data FIFO is empty	R
5	CRC_RES_ERR	Response CRC error. 0: No error on the response CRC 1: CRC error occurred on the response	R
4	CRC_READ_ERROR	Read CRC error. 0: No error on received data 1: CRC error occurred on received data	R
3:2	CRC_WRITE_ERROR	Write CRC error. 00: No error on transmission of data 01: Card observed erroneous transmission of data 10: No CRC status is sent back 11: Reserved	R
1	TIME_OUT_RES	Response Time Out. 0: Card response has not timed out 1: Card response has time out	R
0	TIME_OUT_READ	Read time out. 0: Card read data has not timed out 1: Card read data has timed out	R

24.7.2.3 MSC Clock Rate Register (MSC_CLKRT,0x08)

The MSC_CLKRT register specifies the frequency division of the MMC/SD bus clock. The software is responsible for setting this register.

	MSC_CLKRT																BASE + 0x08															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																													CLK_RATE		
RST																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:3	Reserved	Writing has no effect, read as zero.	R
2:0	CLK_RATE	Clock rate. 000: DEV_CLK 001: 1/2 of DEV_CLK 010: 1/4 of DEV_CLK 011: 1/8 of DEV_CLK 100: 1/16 of DEV_CLK 101: 1/32 of DEV_CLK 110: 1/64 of DEV_CLK 111: 1/128 of DEV_CLK This field must be set to 0 when the controller works during normal writing or reading.	RW

24.7.2.4 MSC Command and Data Control Register (MSC_CMDAT,0x0C)

	MSC_CMDAT																BASE + 0x0C															
Bit	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
	CCS_EXPECTED	READ_CEATA	Reserved		DIS_BOOT	Reserved_0	EXP_BOOT_ACK	BOOT_MODE	Reserved					Reserved_0	SDIO_PDRT	AUTO_CMD12	RTRG		TTRG		IO_ABORT	BUS_WIDTH		Reserved	INIT	BUSY	STREAM_BLOCK	WRITE_READ	DATA_EN	RESPONSE_FORMAT		
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	CCS_EXPECTED	0: interrupts are not enabled in CE-ATA device, or commands does not expect CCS from device 1: interrupts are enabled in CE_ATA device, or RW_BLK command expects command completion signal from device If the command expects Command Completion Signal	RW

		(CCS) from the device, the software should set the control bit. It is auto cleared 0 by hardware.	
30	READ_CEATA	0: host is not performing read access (RW_BLK or RW_REG) towards CE_ATA device 1: host id performing read access (RW_BLK or RW_REG) towards CE_ATA device Software should set the bit to indicate that CE_ATA device is being accessed for read transfer. The bit is used to disable read data timeout indication while performing CE_ATA read transfers. It is auto cleared 0 by hardware.	RW
29:28	Reserved	Writing has no effect, read as zero.	R
27	DIS_BOOT	0: Do nothing 1: Stop boot operation DIS_BOOT should not be used with ENA_BOOT at the same time.	RW
26	Reserved_0	Inner usage, keep to 0.	RW
25	EXP_BOOT_ACK	Whether boot acknowledge pattern is expected or not 0: Boot acknowledge pattern is not expected 1: Boot acknowledge pattern is expected	RW
24	BOOT_MODE	Boot mode operation selection. 0: Ignore. 1: Alternative boot operation	RW
23:19	Reserved	Writing has no effect, read as zero.	R
18	Reserved_0	Inner usage, keep to 0.	RW
17	SDIO_PRDT	Determine whether SDIO interrupt is 2 cycle or extend more cycle when data block last is transferred. 0: more cycle (like single block) 1: exact 2 cycle	RW
16	AUTO_CMD12	This field controls use of auto command functions. 0: Auto CMD12 is disabled 1: MSC controller will issue CMD12 automatically after the last data block transfer is finished. when stop command has finished, it is auto cleared 0 by hardware.	RW
15:14	RTRG	Receive FIFO Trigger Value Select. These bits set the receive FIFO half-empty threshold value, when the number of transmit FIFO >= threshold value , RXFIFO_RD_REQ will be set to 1. 00: more than or equal to 16 01: more than or equal to 32 10: more than or equal to 64	RW

		11: more than or equal to 96	
13:12	TTRG	Transmit FIFO Trigger Value Select. These bits set the transmit FIFO half-empty threshold value, when the number of transmit FIFO < threshold value , TXFIFO_WR_REQ will be set to 1. 00: less than 16 01: less than 32 10: less than 64 11: less than 96	RW
11	IO_ABORT	Specifies the current command is used to abort data transfer. 0: Nothing 1: The current command is used to abort transfer It is auto cleared 0 by hardware.	WR
10:9	BUS_WIDTH	Specifies the width of the data bus. 00: 1-bit 01: Reserved 10: 4-bit 11: 8bit	WR
8	Reserved	Writing has no effect. Read as zero.	R
7	INIT	80 initialization clocks. 0: Issues command directly 1: Issues 80 clocks before command	W
6	BUSY	Specifies whether a busy signal is expected after the current command. This bit is for no data command/response transactions only. 0: Not expect a busy signal 1: Expects a busy signal. If the response is R1b, then set it	WR
5	STREAM_BLOCK	Stream mode. 0: Data transfer of the current command sequence is not in stream mode 1: Data transfer of the current command sequence is in stream mode	WR
4	WRITE_READ	Data Transfer Direction Selection. 0: Read (from card to host) 1: Write (from host to card)	RW
3	DATA_EN	Specifies whether the current command includes data transfer or not. It is also used to reset RX_FIFO and TX_FIFO. 0: Current command without data transfer 1: Current command with data transfer	RW
2:0	RESPONSE_FORMAT	Response Type Selection.	RW

		000: No response 001: Format R1 and R1b 010: Format R2 011: Format R3 100: Format R4 101: Format R5 110: Format R6 111: Format R7	

24.7.2.5 MSC Response Time Out Register (MSC_RESTO,0x10)

	MSC_RESTO																BASE + 0x10															
Bit																	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	RES_TO															
RST																	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:0	RES_TO	Specifies the maximum number of MSC_CLK clock cycles between the end bit of the command and the response from the SD card. The default value is 128.	RW

24.7.2.6 MSC Read Time Out Register (MSC_RDTO,0x14)

	MSC_RDTO																																BASE + 0x14
Bit	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
	READ_TO																																
RST	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Name	Description	RW
31:0	READ_TO	Specifies the maximum number of clocks between the command and when the MMC/SD host controller turns on the time-out error for the received data. The unit is MSC_CLK.	RW

24.7.2.7 MSC Block Size Register (MSC_BLKLEN,0x18)

	MSC_BLKLEN															BASE + 0x18															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																BLK_LEN															
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:0	BLK_LEN	Specifies the number of bytes in a block, and is normally set to 0x200 for MMC/SD data transactions. The value Specified in the cards CSD.	RW

24.7.2.8 MSC Block Count Register (MSC_NOB,0x1C)

	MSC_NOB															BASE + 0x1C															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																NOB															
RST																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:0	NOB	Specifies the number of blocks in a data transfer. One block is a possibility.	RW

24.7.2.9 MSC Successfully-transferred Blocks Count Register (MSC_SNOB,0x20)

In block mode, the MSC_SNOB register records the number of successfully transferred blocks. If the last block has CRC error, this register also summaries it. It is used to query blocks for multiple block transfer.

	MSC_SNOB															BASE + 0x20															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																MSC_SNOB															
RST																?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
15:0	MSC_SNOB	Specify the number of successfully transferred blocks for a multiple block transfer.	R

24.7.2.10 MSC Interrupt Mask Register (MSC_IMASK,0x24)

	MSC_IMASK																BASE + 0x24															
Bit	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
	DMA_DATA_DONE	Reserved_1		PIN_LEVEL					WR_ALL_DONE	Reserved		BCE	BDE	BAE	BAR	DMAEND	AUTO_CMD12_DONE	DATA_FIFO_FULL	DATA_FIFO_EMP	CRC_RES_ERR	CRC_READ_ERR	CRC_WRITE_ERR	TIME_OUT_RES	TIME_OUT_READ	SDIO	TXFIFO_WR_REQ	RXFIFO_RD_REQ	Reserved		END_CMD_RES	PRG_DONE	DATA_TRAN_DONE
RST	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	Name	Description	RW
31	DMA_DATA_DONE	Mask the interrupt DMA_DATA_DONE 0: Not masked 1: Masked	RW
30:29	Reserved_1	Inner usage, keep to 1.	RW
28:24	PIN_LEVEL	Mask the interrupt of PIN_LEVEL separately 0: Not masked 1: Masked	RW
23	WR_ALL_DONE	Mask the interrupt WR_ALL_DONE 0: Not masked 1: Masked	RW
22:21	Reserved	Writing has no effect, read as zero.	R
20	BCE	Boot CRC error. 0: Not masked 1: Masked	RW
19	BDE	Boot data end. 0: Not masked 1: Masked	RW
18	BAE	Mask the interrupt of BAE. 0: Not masked. 1: Masked.	RW
17	BAR	Mask the interrupt of BAR. 0: Not masked. 1: Masked.	RW
16	DMAEND	Mask the interrupt of DMA end. 0: Not masked. 1: Masked.	RW
15	AUTO_CMD12_DONE	Mask the interrupt AUTO_CMD12_DONE. 0: Not masked 1: Masked	RW
14	DATA_FIFO_FULL	0: Not masked	RW

		1: Masked	
13	DATA_FIFO_EMP	0: Not masked 1: Masked	RW
12	CRC_RES_ERR	0: Not masked 1: Masked	RW
11	CRC_READ_ERR	0: Not masked 1: Masked	RW
10	CRC_WRITE_ERR	0: Not masked 1: Masked	RW
9	TIME_OUT_RES	0: Not masked 1: Masked	RW
8	TIME_OUT_READ	0: Not masked 1: Masked	RW
7	SDIO	Mask the interrupt from the SD I/O card. 0: Not masked 1: Masked	WR
6	TXFIFO_WR_REQ	Mask the Transmit FIFO write request interrupt. 0: Not masked 1: Masked	WR
5	RXFIFO_RD_REQ	Mask the Receive FIFO read request interrupt. 0: Not masked 1: Masked	WR
4:3	Reserved	Writing has no effect, read as zero.	R
2	END_CMD_RES	Mask the End command response interrupt. 0: Not masked 1: Masked	WR
1	PRG_DONE	Mask the Programming done interrupt. 0: Not masked 1: Masked	WR
0	DATA_TRAN_DONE	Mask the Data transfer done interrupt. 0: Not masked 1: Masked	WR

24.7.2.11 MSC Interrupt Flag Register (MSC_IFLG,0x28)

The MSC_IFLG register shows the currently requested interrupt. The FIFO request interrupts, TXFIFO_WR_REQ, and RXFIFO_RD_REQ are masked off with the DMA function is used. The software is responsible for monitoring these bit in program I/O mode.

	MSC_IFLG																BASE + 0x28													
Bit	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
	DMA_DATA_DONE	Reserved_0		PIN_LEVEL					WR_ALL_DONE	Reserved		BCE	BDE	BAE	BAR	DMAEND	AUTO_CMD12_DONE	DATA_FIFO_FULL	DATA_FIFO_EMP	CRC_RES_ERR	CRC_READ_ERR	CRC_WRITE_ERR	TIME_OUT_RES	TIME_OUT_READ	SDIO	TXFIFO_WR_REQ	RXFIFO_RD_REQ	Reserved		END_CMD_RES	PRG_DONE	DATA_TRAN_DONE
RST	0	0	0	?	?	?	?	?	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31	DMA_DATA_DONE	This field will be set to 1 when DATA_TRAN_DONE and DMAEND are valid. 0: the interrupt is not detected 1: the interrupt is detected	RW
30:29	Reserved_0	Inner usage, keep to 0.	RW
28:24	PIN_LEVEL	Indicates PIN_LEVEL interrupt. 0: the interrupt is not detected 1: the interrupt is detected Write 1 to clear.	RWC
23	WR_ALL_DONE	This field will be set to 1 when DATA_TRAN_DONE, DMAEND and PRG_DONE are valid. 0: the interrupt is not detected 1: the interrupt is detected	RW
22:21	Reserved_0	Inner usage, keep to 0	R
20	BCE	Boot CRC error. 0: the interrupt is not detected 1: the interrupt is detected Write 1 to clear.	RWC
19	BDE	Boot data end. 0: the interrupt is not detected 1: the interrupt is detected Write 1 to clear.	RWC
18	BAE	Indicates the BAE interrupt. 0: the interrupt is not detected 1: the interrupt is detected Write 1 to clear.	RWC
17	BAR	Indicates the BAR interrupt. 0: the interrupt is not detected 1: the interrupt is detected Write 1 to clear.	RWC
16	DMAEND	Indicates the DMA end interrupt.	RWC

		0: the interrupt is not detected 1: the interrupt is detected Write 1 to clear.	
15	AUTO_CMD12_DONE	Indicates AUTO_CMD12_DONE interrupt. 0: the interrupt is not detected 1: the interrupt is detected Write 1 to clear.	RWC
14	DATA_FIFO_FULL	Indicate data FIFO is full interrupt. 0: the interrupt is not detected 1: the interrupt is detected	R
13	DATA_FIFO_EMP	Indicate data FIFO is empty interrupt. 0: the interrupt is not detected 1: the interrupt is detected	R
12	CRC_RES_ERR	Indicate response CRC error interrupt. 0: the interrupt is not detected 1: the interrupt is detected Write 1 to clear	RWC
11	CRC_READ_ERR	Indicate CRC read error interrupt. 0: the interrupt is not detected 1: the interrupt is detected Write 1 to clear	RWC
10	CRC_WRITE_ERR	Indicate CRC write error interrupt. 0: the interrupt is not detected 1: the interrupt is detected Write 1 to clear	RWC
9	TIME_OUT_RES	Indicate response time out interrupt. 0: the interrupt is not detected 1: the interrupt is detected Write 1 to clear	RWC
8	TIME_OUT_READ	Indicate read time out interrupt. 0: the interrupt is not detected 1: the interrupt is detected Write 1 to clear	RWC
7	SDIO	Indicates whether the interrupt from SDIO is detected. 0: The interrupt from SDIO is not detected 1: The interrupt from SDIO is detected	R
6	TXFIFO_WR_REQ	Transmit FIFO write request. Set if data FIFO becomes half empty. (the number of words is < 8) 0: No Request for data Write to MSC_TXFIFO 1: Request for data write to MSC_TXFIFO	R
5	RXFIFO_RD_REQ	Receive FIFO read request. Set if data FIFO becomes half full (the number of words is >= 8) or the entries in data FIFO are the last read data.	R

		0: No Request for data read from MSC_RXFIFO 1: Request for data read from MSC_RXFIFO	
4:3	Reserved	Writing has no effect, read as zero.	R
2	END_CMD_RES	Indicates whether the command/response sequence has been finished. 0: The command/response sequence has not been finished 1: The command/response sequence has been finished Write 1 to clear.	RWC
1	PRG_DONE	Indicates whether card has finished programming. 0: Card has not finished programming and is busy 1: Card has finished programming and is no longer busy Write 1 to clear.	RWC
0	DATA_TRAN_DONE	Indicates whether data transfer is done. Note that for stream read/write, only when CMD12 (STOP_TRANS) has been sent, is this bit set. 0: Data transfer is not complete 1: Data transfer has completed or an error has occurred Write 1 to clear.	RWC

24.7.2.12 MSC Command Index Register (MSC_CMD,0x2C)

	MSC_CMD																BASE + 0x2C							
Bit																	7	6	5	4	3	2	1	0
																	Reserved		CMD_INDEX					
RST																	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:6	Reserved	Writing has no effect, read as zero.	R
5:0	CMD_INDEX	Specifies the command index to be executed.	RW

24.7.2.13 MSC Command Argument Register (MSC_ARG,0x30)

	MSC_ARG																BASE + 0x30															
Bit	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
	ARG																															
RST	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

Bits	Name	Description	RW
31:0	ARG	Specifies the argument for the current command.	RW

24.7.2.14 MSC Response FIFO Register (MSC_RES,0x34)

The read-only MMC/SD Response FIFO register (RES_FIFO) holds the response sent back to the MMC/SD controller after every command. The size of this FIFO is 8 x 16-bit. The RES_FIFO does not contain the 7-bit CRC for the response. The Status for CRC checking and response time-out status is in the status register MSC_STAT.

The first half-word read from the response FIFO is the most significant half-word of the received response.

	MSC_RES															BASE + 0x34															
Bit																15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																DATA															
RST																?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
15:0	DATA	Contains the response to the command that is sent by the MMC/SD controller.	R

24.7.2.15 MSC Receive FIFO Port Register (MSC_RXFIFO,0x38)

The MSC_RXFIFO is used to read the data from a card. It is read-only to the software, and is read on 32-bit boundary. The size of this FIFO is 128 x 32-bit.

	MSC_RXFIFO																BASE + 0x38															
Bit	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
	DATA																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:0	DATA	One word of read data.	R

24.7.2.16 MSC Transmit FIFO Port Register (MSC_TXFIFO,0x3C)

The MSC_TXFIFO is used to write the data to a card. It is write-only to the software, and is written on 32-bit boundary. The size of this FIFO is 128 x 32-bit.

	MSC_TXFIFO																BASE + 0x3C													
Bit	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
	DATA																															
RST	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

Bits	Name	Description	RW
31:0	DATA	One word of write data.	W

24.7.2.17 MSC Low Power Mode Register (MSC_LPM,0x40)

The MSC_LPM is used to control whether MSC controller enters Low-Power Mode.

	MSC_LPM																BASE + 0x40													
Bit	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
	DRV_SEL		SMP_SEL	Reserved																												LPM
RST	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

Bits	Name	Description	RW
31:30	DRV_SEL	Drive clock selection 0: CMD and data are driven by MSC_CLK falling edge 1: CMD and data are driven by clock rising edge which is 1ns delayed by MSC_CLK 2: CMD and data are driven by clock rising edge which is 1/4 phase delayed by MSC_CLK 3: Reserved When this field is not zero, MSC_CLKRT.CLK_RATE must be set to 0.	RW
29	SMP_SEL	Sample clock selection 0: CMD and data are sampled by MSC_CLK rising edge 1: CMD and data are sampled by clock rising edge which is 1/4 or 1/2 phase delayed by MSC_CLK. When this field is 1, MSC_CLKRT.CLK_RATE must be set to 0.	
28:1	Reserved	Writing has no effect, read as zero.	R
0	LPM	Low Power Mode Enable 0: Disable low power mode 1: Enable low power mode Clock will stop when card in idle (should be normally set to only MMC and SD cards. For SDIO cards, if interrupts must be detected, clock should not be stopped)	RW

		When software sets the bit, MSC clock can auto be stopped. **NOTE:** when set the bit, the clock start and stop can be not use.	

24.7.2.18 MSC DMA Control Register (MSC_DMAC,0x44)

The MSC_DMAC register is used to control the DMA transfer.

	MSC_DMAC																BASE + 0x44													
Bit	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
	Reserved																								MODE_SEL	AOFST		ALIGNEN	INCR		DMASEL	DMAEN
RST	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

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect. Read as zero.	R
7	MODE_SEL	Transfer mode selection. 0: Do not specify the transfer length. This is the standard transfer. 1: Specify the transfer length. If this mode is selected, the transfer performance can be improved.	
6:5	AOFST	Address Offset. This field is effective only when ALIGNEN is 1b. 00b:The lowest 2 bit of the data transfer start address is 00b 01b: The lowest 2 bit of the data transfer start address is 01b 10b: The lowest 2 bit of the data transfer start address is 10b 11b: The lowest 2 bit of the data transfer start address is 11b	RW
4	ALIGNEN	Align Enable 0: Only word boundary data transfer is supported 1: Byte boundary data transfer is supported	RW
3:2	INCR	Burst type selection. The field only effects to special DMA in MSC controller. 2'b00: The burst type of DMA is INCR16. 2'b01: The burst type of DMA is INCR32. 2'b10: The burst type of DMA is INCR64.	RW
1	DMASEL	One of supported DMA modes is selected. 0: Special DMA in MSC controller is used 1: Common DMA in DMAC controller is used	RW
0	DMAEN	DMA Function Enable 0: Disable DMA function 1:Enable DMA function	RW

24.7.2.19 MSC DMA Descriptor Address Register (MSC_DMANDA,0x48)

	MSC_DMANDA																BASE + 0x48															
Bit	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
	NDA																															
RST	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

Bits	Name	Description	RW
31:0	NDA	SDMA Next Descriptor Physical Address The SDMA will change the register when a descriptor is read in. It should be 4-word aligned.	RW

24.7.2.20 MSC DMA Data Address Register (MSC_DMADA,0x4C)

	MSC_DMADA																BASE + 0x4C															
Bit	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
	DA																															
RST	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

Bits	Name	Description	RW
31:0	DA	SDMA Data Physical Address The data address of the current descriptor will be copied to this field. The SDMA will increment it during the data transfer automatically.	R

24.7.2.21 MSC DMA Data Length Register (MSC_DMALEN,0x50)

	MSC_DMALEN																BASE + 0x50															
Bit	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
	LEN																															
RST	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

Bits	Name	Description	RW
31:0	LEN	SDMA Data Length to be Transferred Its unit is byte. The data length of the current descriptor will be copied to this field. The SDMA will decrement it during the data transfer automatically.	R

24.7.2.22 MSC DMA Command Register (MSC_DMACMD,0x54)

	MSC_DMACMD																BASE + 0x54													
Bit	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
	IDI								ID								Reserved					OFFSET		ALIGN_EN						ENDI	LINK
RST	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

Bits	Name	Description	RW
31:24	IDI	Identification of the Interrupt DMA. This field will copy the ID when a DMA transfer interrupt occurs. If the interrupt of DMA is disabled, the field will not change.	R
23:16	ID	Identification of Current DMA Transfer.	R
15:11	Reserved	It has no use in the current version.	R
10:9	OFFSET	Address Offset. This field is effective when ALIGN_EN is set to 1b. 00: data address[1:0] is 00b 01: data address[1:0] is 01b 10: data address[1:0] is 10b 11: data address[1:0] is 11b	R
8	ALIGN_EN	Address Align Enable. This bit should be set to 1b if the data address is not word-aligned.	R
7:2	Reserved	It has no use in the current version.	R
1	ENDI	Interrupt Enable for Current DMA Transfer End 0: Disable interrupt 1: Enable interrupt	R
0	LINK	Control the end of DMA Descriptor. 0: DMA will go to idle state after the current transmission is finished. The MSC_STAT.DMAEND will be set to 1b when the current read/write is finished. If the interrupt mask bit MSC_IMASK.DMAEND is 0b, the interrupt flag MSC_IFLG. DMAEND will be set to 1b also. 1: DMA will continue to fetch another descriptor.	R

24.7.2.23 MSC Control 2 Register (MSC_CTRL2,0x58)

	MSC_CTRL2																BASE + 0x58															
Bit	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
	Reserved			PIP					Reserved_1	Reserved																		STPRM	Reserved_0	SMS		
RST	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:29	Reserved	Writing has no effect, read as zero.	R
28:24	PIP	Pin Level Interrupt Polarity 1: Interrupt can be trigged when pin level is high 0: Interrupt can be trigged when pin level is low	RW
23	Reserved_1	Inner usage, keep to 1.	RW
22:5	Reserved	Writing has no effect, read as zero.	R
4	STPRM	Stop Read Operation Mode Selection 0: Host can stop read operation during data transfer 1: Host only can stop read operation during block gap	RW
3	Reserved_0	Inner usage, keep to 0.	RW
2:0	SMS	Speed Mode Selection 000: Default speed 001: High speed 010: SDR12 011: SDR25 100: SDR50 Others: reserved	RW

24.7.2.24 MSC RTFIFO Data Counter Register(MSC_RTCNT,0x5C)

	MSC_RTCNT																BASE + 0x5C															
Bit	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
	RTCNT																															
RST	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

Bits	Name	Description	RW
31:0	RTCNT	This field indicates how many data in word units are stored in RTFIFO.	R

24.8 Operation Flow

24.8.1 Data FIFOs

The controller FIFOs for the response tokens, received data, and transmitted data are MSC_RES, MSC_RXFIFO, and MSC_TXFIFO, respectively. These FIFOs are accessible by the software and are described in the following paragraphs.

24.8.1.1 Response FIFO (MSC_RES)

The response FIFO, MSC_RES, contains the response received from an MMC/SD card after a command is sent from the controller. MSC_RES is a read-only, 16-bit, and 8-entry deep FIFO.
The FIFO will hold all possible response lengths. Responses that are only one byte long are located on the LSB of the 16-bit entry in the FIFO. The first half-word read from the response FIFO is the most significant half-word of the received response. For example, if the response format is R1, then the response read from RES_FIFO is bit [47:32], bit[31:16], bit[15:0] and in the third half-word only the low 8-bit is effective response [15:8] and the high 8-bit is ignored. If the response format is R2, then the response read from MSC_RES is bit [135:8] and needs reading 8 times.
The FIFO does not contain the response CRC. The status of the CRC check is in the status register, MSC_STAT.

24.8.1.2 Receive/Transmit Data FIFO (MSC_RXFIFO/MSC_TXFIFO)

The receive data FIFO and transmit data FIFO share one 16-entry x 32-bit FIFO, because at one time data are only received or are only transmitted. If it is used to receive data, it is called MSC_RXFIFO and read-only. If it is used to transmit data, it is called MSC_TXFIFO and write-only.
Data FIFO and its controls are cleared to a starting state after a system reset or at the beginning of the operations which include data transfer. (MSC_CMDAT[DATA_EN] == 1)
If at any time MSC_RXFIFO becomes full and the data transmission is not complete, the controller turns the MSC_CLK off to prevent any overflows. When the clock is off, data transmission from the card stops until the clock is turned back on. After MSC_RXFIFO is not full, the controller turns the clock on to continue data transmission. The full status of the FIFO is registered in the MSC_STAT [DATA_FIFO_FULL] bit.
If at any time MSC_TXFIFO becomes empty and the data transmission is not complete, the controller turns the MSC_CLK off to prevent any under run. When the clock is off, data transmission to the card stops until the clock is turned back on. When MSC_TXFIFO is no longer empty, the controller automatically restarts the clock. The empty status of the FIFO is registered in the MSC_STAT [DATA_FIFO_EMPTY] bit.
The FIFO is readable on word (32-bit) boundaries. The max read/written number is 16 words. The controller can correctly process big-endian and little-endian data.
Because at the beginning of the operation which include data transfer (MSC_CMDAT [DATA_EN] == 1), Data FIFO and its controls are cleared, software should guarantee data in FIFO have been read/written before beginning a new command.

24.8.2 DMA and Program I/O

Software may communicate to the MSC controller via the DMA or program I/O. The SDMA and CDMA

are supported at the same time. SDMA is the special DMA in MSC controller, while CDMA is the common DMA in DMAC controller.

24.8.2.1 Structure of SDMA Descriptor

SDMA is a descriptor DMA. And a descriptor can be linked to another one by the NDA until the LINK in the 4th word is set to 1b.

A SDMA descriptor includes 4 words.

Word 1: The physical address of the next SDMA descriptor.

Bits	Name	Description
31:0	NDA	Next descriptor physical address. It should be 4-word aligned.

Word 2: The system memory address where the data will be read from or stored to.

Bits	Name	Description
31:0	DA	Data Address.

Word 3: The length of data to be transferred.

Bits	Name	Description
31:0	LEN	Unit is byte.

Word 4: The command for the current SDMA.

Bits	Name	Description
31:24	Reserved	It has no use in the current version.
23:16	ID	Identification of Current DMA Transfer.
15:2	Reserved	It has no use in the current version.
1	ENDI	Interrupt Enable for Current DMA Transfer End 0: Disable interrupt 1: Enable interrupt
0	LINK	Control the end of DMA Descriptor. 0: DMA will continue to fetch another descriptor. 1: DMA will go to idle state after the current transfer. The MSC_STAT.DMAEND will be set to 1b when the current read/write is finished. If the interrupt mask bit MSC_IMASK.DMAEND is 0b, the interrupt flag MSC_IFLG. DMAEND will be set to 1b also.

24.8.2.2 Operation of SDMA

Step 1: Prepare the descriptor in system memory

Step 2: If the address is not word boundary, it is suggested to configure MSC_DMAC.ALIGNEN and

MSC_DMAC.AOFST

Step 3: Write the descriptor address to MSC_DMANDA

Step 4: Configure MSC controller to read or write operation, e,g, set MSC_NOB, MSC_BLEN and MSC_CMDAT.WRITE_READ etc.

Step 5: If DMAEND interrupt is wanted, clear MSC_IMASK.DMAEND, otherwise, set this bit

Step 6: Select SDMA by setting MSC_DMAC.DMASEL to 0b

Step 7: Start DMA transfer by setting MSC_DMAC.DMAEN

Step 8: Waiting the DMAEND interrupt (if interrupt is used) or status (if interrupt is not used)

Step 9: Disable the SDMA by configuring MSC_DMAC.DMAEN to 0b

24.8.2.3 Operation of CDMA

To access MSC_RXFIFO/MSC_TXFIFO with the DMA, the software must program the DMA to read or write the FIFO with source port width 32-bit, destination port width 32-bit, transfer data size 32-byte, transfer mode single. For example, to write 64 bytes of data to the MSC_TXFIFO, the software must program the DMA as follows:

```
DMA_DCTRn = 2                // Write 2 32-bytes (64 bytes)
DMA_DCCRn[SWDH] = 0          // source port width is 32-bit
DMA_DCCRn[DWDH] = 0          // destination port width is 32-bit
DMA_DCCRn[DS] = 4            // transfer data size is 32-byte
DMA_DCCRn[TM] = 4            // transfer mode is single
DMA_DCCRn[RDIL] = 0          // request detection interval length is 0
```

The number of 32-bytes should be calculated from the number of transferred bytes as follows:

The number of words = (The number of bytes + 31) / 32

If the number of transferred bytes is not the multiple of 4, the controller can correctly process endian. The DMA trigger level is 8 words, that is to say, the DMA read trigger is when data words in MSC_RXFIFO is >= 8 and the DMA write trigger is when data words in MSC_TXFIFO is < 8. Software can also configure DMA registers based on requirements, but the above 32-byte transfer data size is most efficient.

24.8.2.4 Operation of Program I/O

With program I/O, the software waits for the MSC_IFLG [RXFIFO_RD_REQ] or MSC_IFLG [TXFIFO_WR_REQ] interrupts before reading or writing the respective FIFO.

NOTES:

1. The MSC_CMDAT [DMA_EN] bit must be set to a 1 to enable communication with the DMA and it must be set to a 0 to enable program I/O.
2. DMA can be enabled only after MSC_CMDAT is written, because MSC_CMDAT [DATA_EN] is used to reset TX/RXFIFO.

24.8.3 Start and Stop clock

The software stops the clock as follows:

1. Write MSC_CTRL with 0x01 to stop the MMC/SD bus clock.
2. Wait until MSC_STAT[CLK_EN] becomes zero.

To start the clock the software writes MSC_CTRL with 0x02.

24.8.4 Software Reset

Reset includes the MSC reset and the card reset.

The MSC reset is through MSC_CTRL [RESET] bit.

The card reset is to make the card into idle state. CMD0 (GO_IDLE_STATE) sets the MMC and SD memory cards into idle state. CMD52 (IO_RW_DIRECT, with argument 0x88000C08) reset the SD I/O card. The MMC/SD card are initialized with a default relative card address (RCA = 0x0001 for MMC and RCA = 0x0000 for SD) and with a default driver stage register setting (lowest speed, highest driving current capability).

The following registers must be set before the clock is started:

Step 1. Stop the clock.
Step 2. Set MSC_CTRL register to 0x08 to reset MSC.
Step 3. Wait while MSC_STAT [IS_RESETTING] is 1.
Step 4. Set MSC_CMD with CMD0.
Step 5. Update the MSC_CMDAT register as follows:
  a Write 0x0000 to MSC_CMDAT [RESPONSE_FORMAT].
  b Clear the MSC_CMDAT [DATA_EN] bit.
  c Clear the MSC_CMDAT [BUSY] bit.
  d Clear the MSC_CMDAT [INIT] bit.
Step 6. Start the clock.
Step 7. Start the operation. (write MSC_CTRL with 0x04)
Step 8. Wait for the END_CMD_RES interrupt.
Step 9. Set MSC_CMD with CMD52.
Step 10. Set MSC_ARG with 0x88000C08.
Step 11. Update the MSC_CMDAT register as follows:
  a Write 0x005 to MSC_CMDAT [RESPONSE_FORMAT].
  b Clear the MSC_CMDAT [DATA_EN] bit.
  c Clear the MSC_CMDAT [BUSY] bit.
  d Clear the MSC_CMDAT [INIT] bit.
Step 12. Start the operation.
Step 13. Wait for the END_CMD_RES interrupt.

24.8.5 Voltage Validation and Card Registry

At most 10 MMC and 1 SD (either SDMEM or SDIO) can be inserted MMC/SD bus at the same time, and their voltage validation and card registry steps are different, so the software should be programmed as follows:

Step 1. Check whether SDIO card is inserted.
Step 2. Check whether SDMEM card is inserted.
Step 3. Check whether MMC cards are inserted.

24.8.5.1 Check SDIO

The commands are sent as follows:

Step 1. (Optional) Send CMD52 (IO_RW_DIRECT) with argument 0x88000C08 to reset SDIO card.
Step 2. Send CMD5 (IO_SEND_OP_CMD) to validate voltage.
Step 3. If the response is correct and the number of IO functions > 0, then continue, else go to check SDMEM.
Step 4. If C-bit in the response is ready (the initialization has finished), go to 6.
Step 5. Send CMD5 (IO_SEND_OP_CMD) to validate voltage, then go to 4.
Step 6. If memory-present-bit in the response is true, then it is a combo card (SDIO + Memory), else it is only a SDIO card.
Step 7. If it is a combo card, go to check SDMEM to initialize the memory part.
Step 8. Send CMD3 (SET_RELATIVE_ADDR) to let the card publish a RCA. The RCA is returned from the response.
Step 9. If do not accept the new RCA, go to 8, else record the new RCA.
Step 10. Go to check MMC, because we can assure that there is no SDMEM card.

24.8.5.2 Check SDMEM

If there is no SDIO card or there is a combo card, continue to check SDMEM.

The commands are sent as follows:

Step 1. (Optional) Send CMD0 (GO_IDLE_STATE) to reset MMC and SDMEM card. This command has no response.
Step 2. Send CMD55. Here the default RCA 0x0000 is used for CMD55.
Step 3. If the response is correct (CMD55 has response), then continue, else go to check MMC.
Step 4. Send ACMD41 (SD_SEND_OP_CMD) to validate voltage (the general OCR value is 0x00FF8000).
Step 5. If the initialization has finished, go to 7. (The response is the OCR register and it includes a status information bit (bit [31]). This status bit is set if the card power up procedure has been finished. As long as the card is busy, the corresponding bit[31] is set to LOW.)
Step 6. Send CMD55 and ACMD41 to validate voltage, and then go to 5.
Step 7. Send CMD2 (ALL_SEND_CID) to get the card CID.
Step 8. Send CMD3 (SET_RELATIVE_ADDR) to let card publish a RCA. The RCA is returned from the response.
Step 9. If do not accept the new RCA, go to 8, else record the new RCA.
Step 10. Go to check MMC.

24.8.5.3 Check MMC

Because there may be several MMC card, so some steps (5 ~ 8) should be repeated several times.

The commands are sent as follows:

Step 1. Send CMD1 (SEND_OP_CMD) to validate voltage (the general OCR value is 0x00FF88000).

Step 2. If the response is correct, then continue, else go to 9.
Step 3. If the initialization has finished, go to 5. (The response is the OCR register and it includes a status information bit (bit [31]). This status bit is set if the card power up procedure has been finished. As long as the card is busy, the corresponding bit[31] is set to LOW.)
Step 4. Send CMD1 (SEND_OP_CMD) to validate voltage, and then go to 3.
Step 5. Send CMD2 (ALL_SEND_CID) to get the card CID.
Step 6. If the response timeout occurs, go to 9.
Step 7. Send CMD3 (SET_RELATIVE_ADDR) to assign the card a RCA.
Step 8. If there are other MMC cards, then go to 5.
Step 9. Finish.

24.8.6 Single Data Block Write

In a single block write command, the following registers must be set before the operation is started:

Step 1. Set MSC_NOB register to 0x0001.
Step 2. Set MSC_BLKLEN to the number of bytes per block.
Step 3. Update the MSC_CMDAT register as follows:
  a Write 0x001 to MSC_CMDAT [RESPONSE_FORMAT].
  b Write 0x2 to MSC_CMDAT [BUS_WIDTH] if the card is SD, else clear it.
  c Set the MSC_CMDAT [DATA_EN] bit.
  d Set the MSC_CMDAT [WRITE_READ] bit.
  e Clear the MSC_CMDAT [STREAM_BLOCK] bit.
  f Clear the MSC_CMDAT [BUSY] bit.
  g Clear the MSC_CMDAT [INIT] bit.
Step 4. Start the operation.
Step 5. Write MSC_IMASK with some value to unmask the expected interrupts.

Then the software must perform the following steps:

Step 1. Wait for the MSC_IFLG [END_CMD_RES] interrupt.
Step 2. Wait for the MSC_IFLG [DATA_TRAN_DONE] interrupt.
  At the same time write data to the MSC_TXFIFO and continue until all of the data have been written to the FIFO.
Step 3. Wait for MSC_IFLG [PROG_DONE] interrupt. This interrupt indicates that the card has finished programming. Certainly software may start another command sequence on a different card.
Step 4. Read the MSC_STAT register to verify the status of the transaction (i.e. CRC error status).

To address a different card, the software sends a select command to that card by sending a basic no data command and response transaction. To address the same card, the software must wait for MSC_IFLG [PROG_DONE] interrupt. This ensures that the card is not in the busy state.

In addition, CMD26 (PROGRAM_CID), CMD27 (PROGRAM_CSD), CMD42 (LOCK/UNLOCK), CMD56 (GEN_CMD: write) and CMD53 (single block write) operations are similar to single block write.

24.8.7 Single Block Read

In a single block read command, the following registers must be set before the operation is started:

Step 1. Set MSC_NOB register to 0x0001.

Step 2. Set MSC_BLKLEN register to the number of bytes per block.

Step 3. Update the following bits in the MSC_CMDAT register:

a Write 0x001 to MSC_CMDAT [RESPONSE_FORMAT].

b Write 0x2 to MSC_CMDAT [BUS_WIDTH] if the card is SD, else clear it.

c Set the MSC_CMDAT [DATA_EN] bit.

d Clear the MSC_CMDAT [WRITE_READ] bit.

e Clear the MSC_CMDAT [STREAM_BLOCK] bit.

f Clear the MSC_CMDAT [BUSY] bit.

g Clear the MSC_CMDAT [INIT] bit.

Step 4. Start the operation.

Step 5. Write MSC_IMASK with some value to unmask the expected interrupts.

Then the software must perform the following steps:

Step 1. Wait for the MSC_IFLG [END_CMD_RES] interrupt.

Step 2. Wait for the MSC_IFLG [DATA_TRAN_DONE] interrupt.
At the same time read data from the MSC_RXFIFO as data becomes available in the FIFO, and continue reading until all data is read from the FIFO.

Step 3. Read the MSC_STAT register to verify the status of the transaction (i.e. CRC error status).

In addition, CMD30 (SEND_WRITE_PROT), ACMD13 (SD_STATUS), CMD56 (GEN_CMD-read), ACMD51 (SEND_SCR) and CMD53 (single block read) are similar to single block read.

24.8.8 Multiple Block Write

The multiple block write mode is similar to the single block write mode, except that multiple blocks of data are transferred. Each block is the same length. All the registers are set as they are for the single block write, except that the MSC_NOB register is set to the number of blocks to be written.

The multiple block write mode also requires a stop transmission command, CMD12, after the data is transferred to the card. After the MSC_IFLG [DATA_TRAN_DONE] interrupt occurs, the software must program the controller register to send a stop data transmission command.

If multiple block write with pre-defined block count (refer to MMC spec v-3.3) is used, CMD12 should not be sent.

For SDIO card, CMD53 (multiple block write) is also similar, but when IO abort (CMD52) is sent, MSC_CMDAT [IO_ABORT] should be 1.

Table 24-6 How to stop multiple block write

Operation	Stop condition	Software processing
Open-ended or SDIO infinite	After write MSC_NOB blocks	1 Wait for DATA_TARN_DONE interrupt. 2 Send CMD12 or CMD52. (IO abort) 3 Wait for END_CMD_RES and PRG_DONE interrupt.
Open-ended or SDIO infinite	Stop writing in advance (not write MSC_NOB blocks)	1 Set MSC_CTRL [EXIT_MULTIPLE]. 2 Wait for DATA_TRAN_DONE interrupt. 3 Send CMD12 or CMD52. (IO abort) 4 Wait for END_CMD_RES and PRG_DONE interrupt.
Predefined block or SDIO finite	After writing MSC_NOB blocks	1 Wait for DATA_TRAN_DONE interrupt.
Predefined block or SDIO finite	Stop writing in advance (not write MSC_NOB blocks)	1 Set MSC_CTRL [EXIT_MULTIPLE]. 2 Wait for DATA_TRAN_DONE interrupt. 3 Send CMD12 or CMD52. (IO abort) 4 Wait for END_CMD_RES and PRG_DONE interrupt.

24.8.9 Multiple Block Read

The multiple blocks read mode is similar to the single block read mode, except that multiple blocks of data are transferred. Each block is the same length. All the registers are set as they are for the single block read, except that the MSC_NOB register is set to the number of blocks to be read.

The multiple blocks read mode requires a stop transmission command, CMD12, after the data from the card is received. After the MSC_IFLG [DATA_TRAN_DONE] interrupt has occurred, the software must program the controller registers to send a stop data transmission command.

If multiple block read with pre-defined block count (refer to MMC spec v-3.3) is used, CMD12 should not be sent.

For SDIO card, CMD53 (multiple block read) is also similar, but when IO abort (CMD52) is sent, MSC_CMDAT [IO_ABORT] should be 1.

Table 24-7 How to stop multiple block read

Operation	Stop condition	Software processing
Open-ended or SDIO infinite	After reading MSC_NOB blocks	1 Wait for DATA_TRAN_DONE interrupt. 2 Send CMD12 or CMD52. (IO abort) 3 Wait for END_CMD_RES interrupt.
Open-ended or SDIO infinite	Stop reading in advance (not write MSC_NOB blocks)	1 Set MSC_CTRL [EXIT_MULTIPLE]. 2 Wait for DATA_TRAN_DONE interrupt. 3 Send CMD12 or CMD52. (IO abort)

		4	Wait for END_CMD_RES interrupt.
Predefined block or SDIO finite	After reading MSC_NOB blocks	1	Wait for DATA_TRAN_DONE interrupt.
Predefined block or SDIO finite	Stop reading in advance (not write MSC_NOB blocks)	1 2 3 4	Set MSC_CTRL [EXIT_MULTIPLE]. Wait for DATA_TRAN_DONE interrupt. Send CMD12 or CMD52. (IO abort) Wait for END_CMD_RES interrupt.

24.8.10 Stream Write (MMC)

In a stream write command, the following registers must be set before the operation is started:

1. Update MSC_CMDAT register as follows:
   a. Write 0x001 to the MSC_CMDAT [RESPONSE_FORMAT].
   b. Clear the MSC_CMDAT [BUS_WIDTH] because only MMC support stream write.
   c. Set the MSC_CMDAT [DATA_EN] bit.
   d. Set the MSC_CMDAT [WRITE_READ] bit.
   e. Set the MSC_CMDAT [STREAM_BLOCK] bit.
   f. Clear the MSC_CMDAT [BUSY] bit.
   g. Clear the MSC_CMDAT [INIT] bit.
2. Start the operation.
3. Write MSC_IMASK with some value to unmask the expected interrupts.

Then the software must perform the following steps:

1. Wait for the MSC_IFLG [END_CMD_RES] interrupt.
2. Write data to the MSC_TXFIFO and continue until all of the data is written to the Data FIFO.
3. Stop clock. Wait until MSC_STAT[CLK_EN] becomes 0. The clock must be stopped.
4. Set the command registers for a stop transaction command (CMD12) and other registers.
5. Start the clock and start the operation.
6. Wait for the MSC_IFLG [END_CMD_ERS] interrupt.
7. Wait for the MSC_IFLG [DATA_TRAN_DONE] interrupt.
8. Wait for the MSC_IFLG [PRG_DONE] interrupt. This interrupt indicates that the card has finished programming. Certainly software may start another command sequence on a different card.
9. Read the MSC_STAT register to verify the status of the transaction.

To address a different card, the software must send a select command to that card by sending a basic no data command and response transaction. To address the same card, the software must wait for MSC_IFLG [PRG_DONE] interrupt. This ensures that the card is not in the busy state.

If partial blocks are allowed (if CSD parameter WRITE_BL_PARTIAL is set) the data stream can start and stop at any address within the card address space, otherwise it shall start and stop only at block boundaries. If WRITE_BL_PARTIAL is not set, 16 more stuff bytes need to be written after the useful

written data, otherwise only write the useful written data.

24.8.11 Stream Read (MMC)

In a stream read command, the following registers must be set before the operation is turned on:

1 Update the MSC_CMDAT register as follows:
   a Write 0x01 to the MSC_CMDAT [RESPONSE_FORMAT].
   b Clear the MSC_CMDAT [BUS_WIDTH] because only MMC support stream read.
   c Clear the MSC_CMDAT [WRITE_READ] bit.
   d Set the MSC_CMDAT [STREAM_BLOCK] bit.
   e Clear the MSC_CMDAT [BUSY] bit.
   f Clear the MSC_CMDAT [INIT] bit.
2 Start the operation.
3 Write MSC_IMASK with some value to unmask the expected interrupts.

Then the software must perform the following steps:

1 Wait for the MSC_IFLG [END_CMD_RES] interrupt.
2 Read data from the MSC_RXFIFO and continue until all of the expected data has been read from the FIFO.
3 Write MSC_CTRL [EXIT_TRANSER] with 1. If MSC_STAT[DATA_FIFO_FULL] is 1, then read MSC_RXFIFO to make it not full. Because if data FIFO is full, MSC_CLK is stopped. Here, the data FIFO contains useless data.
4 Set the command registers for a stop transaction command (CMD12) and send it. There is no need to stop the clock.
5 Wait for the MSC_IFLG [END_CMD_RES].
6 Wait for the MSC_IFLG [DATA_TRAN_DONE] interrupt.
7 Read the MSC_STAT register to verify the status of the transaction.

24.8.12 Erase, Select/Deselect and Stop

For CMD7 (SELECT/DESELECT_CARD), CMD12 (STOP_TRANSMISSION) and CMD38 (ERASE), the following registers must be set before the operation is started:

1 Update the MSC_CMDAT register as follows:
   a Write 0x01 to the MSC_CMDAT [RESPONSE_FORMAT].
   b Clear the MSC_CMDAT [DATA_EN] bit.
   c Clear the MSC_CMDAT [WRITE_READ] bit.
   d Clear the MSC_CMDAT [STREAM_BLOCK] bit.
   e Set the MSC_CMDAT [BUSY] bit.
   f Clear the MSC_CMDAT [INIT] bit.
2 Start the operation.
3 Write MSC_IMASK with some value to unmask the expected interrupts.

Then the software must perform the following steps:

1 Wait for the MSC_IFLG [END_CMD_RES] interrupt.
2 Wait for the MSC_IFLG [PRG_DONE] interrupt. If CMD12 is sent to terminate data read operation, then there is no need to wait for MSC_IFLG [PRG_DONE] interrupt. This interrupt indicates that the card has finished programming. Certainly software may start another command sequence on a different card.

24.8.13 SDIO Suspend/Resume

The actual suspend/resume steps are as follows:

1 During data transfer, send CMD52 to require suspend. BR and RAW flag should be 1.
2 If BS flag in the response is 0, then suspend has been accepted and goto 4.
3 Send CMD52 to query card status. R flag should be 1. Go to 2.
4 Write MSC_CTRL [EXIT_TRANSFER] with 1.
5 Wait for the MSC_IFLG [DATA_TRAN_DONE] interrupt.
6 Read MSC_NOB, MSC_SNOB and etc, save them into variables.
7 Set registers for high priority transfer and start it.
8 Wait until high priority transfer is finished.
9 Restore registers from variables, but MSC_NOB should be (MSC_NOB – MSC_SNOB).
10 Send CMD52 to require resume. FSx should be resumed function number.

24.8.14 SDIO Read Wait

The actual Read Wait steps are as follows:

1 During multiple block read, read MSC_SNOB. If MSC_SNOB is nearby or equal to MSC_NOB, no need to use Read Wait.
2 Write MSC_CTRL [START_READ WAIT] with 1.
3 Wait until MSC_STAT [IS_READ WAIT] becomes 1.
4 Send CMD52 to query card status.
5 Write MSC_CTRL [STOP_READ WAIT] with 1.

24.8.15 Operation and Interrupt

The software can use polling-status method to operate the MMC/SD card, but this is not the proposed method, because its performance is very low. The proposed method is to use interrupt. Generally there are fixed necessary steps to finish each command. The steps are as follows:

1 (Optional) Stop clock. Poll CLK_EN.
2 Fill the registers (MSC_CMD, MSC_CMDAT, MSC_ARG, MSC_CLKRT, and etc).
3 (Optional) Start clock.
4 Start the operation. Wait for the MSC_IFLG [END_CMD_RES] interrupt.
5 Wait for the MSC_IFLG [DATA_TRAN_DONE] interrupt.
6 Send STOP_TRANS (CMD12) or I/O abort (CMD52). Wait for the MSC_IFLG [END_CMD_ERS] interrupt.
7 Wait for the MSC_IFLG [DATA_TRAN_DONE] interrupt.

8 Wait for the MSC_IFLG [PRG_DONE] interrupt.

24.9 Index

Table 24-8 Description of Proprietary Vocabulary

Name	Description
Block	A number of bytes, basic data transfer unit
CID	Card IDentification number register
CRC	Cyclic Redundancy Check
CSD	Card specific Data register
OCR	Operation Conditions Register
Open-drain	A logical interface operation mode. An external resistor or current source is used to pull the interface level to HIGH, the internal transistor pushes it to LOW.
Push-pull	A logical interface operation mode, a complementary pair of transistors is used to push the interface level to High or Low

Table 24-9 The mapping between Commands and Steps

Index	Abbreviation	1	2	3	4	5	6	7	8	Comments
CMD0	GO_IDLE_STATE	Y	Y	Y	Y					
CMD1	SEND_OP_COND	Y	Y	Y	Y					
CMD2	ALL_SEND_CID	Y	Y	Y	Y					
CMD3	SET_RELATIVE_ADDR	Y	Y	Y	Y					
CMD4	SET_DSR	Y	Y	Y	Y					
CMD7	SELECT/DSELECT_CARD	Y	Y	Y	Y				Y	
CMD9	SEND_CID	Y	Y	Y	Y					
CMD10	SEND_CSD	Y	Y	Y	Y					
CMD11	READ_DAT_UNTIL_STOP	Y	Y	Y	Y		Y	Y		
CMD12	STOP_TRANSMISSION	Y	Y	Y	Y				Y	
CMD13	SEND_STATUS	Y	Y	Y	Y					
CMD15	GO_INACTIVE_STATE	Y	Y	Y	Y					
CMD16	SET_BLOCKLEN	Y	Y	Y	Y					
CMD17	READ_SINGLE_BLOCK	Y	Y	Y	Y	Y				
CMD18	READ_MULTIPLE_BLOCK	Y	Y	Y	Y	Y	Y			Open-ended
CMD18	READ_MULTIPLE_BLOCK	Y	Y	Y	Y	Y				Predefine blocks
CMD20	WRITE_DAT_UNTIL_STOP	Y	Y	Y	Y		Y	Y	Y	
CMD23	SET_BLOCK_COUNT	Y	Y	Y	Y					
CMD24	WRITE_SINGLE_BLOCK	Y	Y	Y	Y	Y			Y	
CMD25	WRITE_MULTIPLE_BLOCK	Y	Y	Y	Y	Y	Y		Y	Open-ended
CMD25	WRITE_MULTIPLE_BLOCK	Y	Y	Y	Y	Y			Y	Predefine blocks

CMD26	PROGRAM_CID	Y	Y	Y	Y	Y			Y	
CMD27	PROGRAM_CSD	Y	Y	Y	Y	Y			Y	
CMD28	SET_WRITE_PROT	Y	Y	Y	Y				Y	
CMD29	CLR_WRITE_PROT	Y	Y	Y	Y				Y	
CMD30	SEND_WRITE_PROT	Y	Y	Y	Y	Y				
CMD32	ERASE_WR_BLOCK_START	Y	Y	Y	Y					
CMD33	ERASE_WR_BLOCK_END	Y	Y	Y	Y					
CMD35	ERASE_GROUP_START	Y	Y	Y	Y					
CMD36	ERASE_GROUP_END	Y	Y	Y	Y					
CMD38	ERASE	Y	Y	Y	Y				Y	
CMD39	FAST_IO	Y	Y	Y	Y					
CMD40	GO_IRQ_STATE	Y	Y	Y	Y					
CMD42	LOCK/UNLOCK	Y	Y	Y	Y	Y			Y	
CMD55	APP_CMD	Y	Y	Y	Y					
CMD56	GEN_CMD	Y	Y	Y	Y	Y				Read
CMD56	GEN_CMD	Y	Y	Y	Y	Y			Y	Write
ACMD6	SET_BUS_WIDTH	Y	Y	Y	Y					
ACMD13	SD_STATUS	Y	Y	Y	Y	Y				
ACMD22	SEND_NUM_WR_BLOCKS	Y	Y	Y	Y					
ACMD23	SET_WR_BLOCK_COUNT	Y	Y	Y	Y					
ACMD41	SD_SEND_OP_COND	Y	Y	Y	Y					
ACMD42	SET_CLR_CARD_DETECT	Y	Y	Y	Y					
ACMD51	SEND_SCR	Y	Y	Y	Y	Y				

NOTE: For stream read/write, STOP_CMD is sent after finishing data transfer. For write, STOP_CMD is with the last six bytes. For read, STOP_CMD is sent after receiving data and card sends some data which MSC ignores.

25 OTG Controller

25.1 Overview

This chapter describes the USB On-The-Go (OTG) implemented in the processor.

The Universal Serial Bus (USB) supports serial data exchanges between a host computer and a variety of simultaneously accessible portable peripherals. Many of these portable devices would benefit a lot from being able to communicate to each other over the USB interface. And OTG make this possible. An OTG device can plays the role of both host and device.

Familiarity with the *Universal Serial Bus Specification*, Revision 1.1 and OTG supplement are necessary to fully understand the material contained in this section.

Features:

- Complies with the USB 2.0 standard for high-speed (480 Mbps) functions and with the *On-The-Go* supplement to the USB 2.0 specification
- Operates either as the function controller of a high- /full-speed USB peripheral or as the host/peripheral in point-to-point or multi-point communications with other USB functions
- Supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP)
- UTMI+ Level 3 Transceiver Interface
- Soft connect/disconnect
- Supports up to 8 bidirectional endpoints
- Supports up to 16 host channels
- Supports control, interrupt, ISO and bulk transfer
- Supports INCR4, INCR8, INCR16, INCR, and SINGLE transfers on the AHB Slave interface
- Software-selectable AHB burst type on AHB Master interface in DMA mode

25.2 Block Diagram

25.2.1 Slave-Only mode

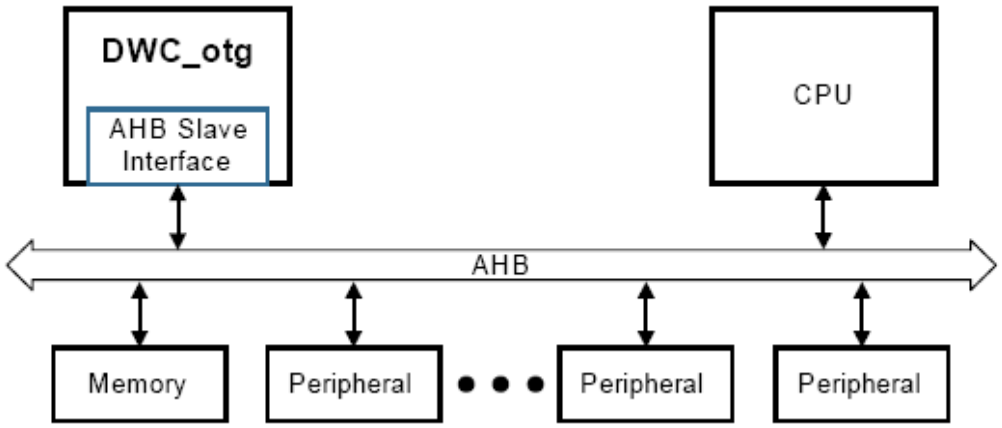


Figure 25-1 OTG Slave-Only mode

25.2.2 Internal DMA mode

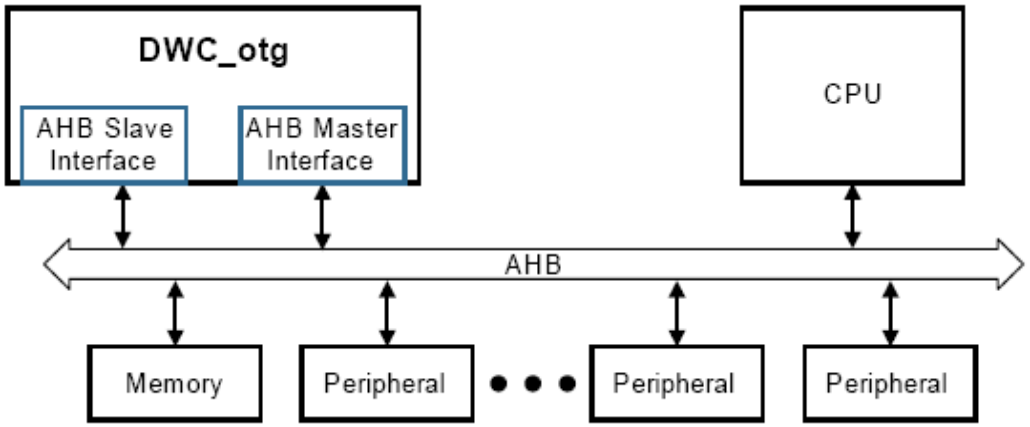


Figure 25-2 OTG Internal DMA mode

25.3 Pin Description

Table 25-1 OTG Pins Description

Name	Type	Description
DP	Inout	Data Positive
DM	Inout	Data Minus
ID	Inout	Identification
DRVVBUS	Out	Charge pump enable

25.4 Register Map

Following chapter will descript the functions of all software accessible registers.

Table 25-2 Registers Memory Map Base Address

Name	Base	Description
OTG	0x13500000	OTG controller base address

Only the Core Global, Data FIFO Access, and Host Port registers can be accessed in both Host and Device modes. When the DWC_otg core is operating in one mode, either Device or Host, the application must not access registers from the other mode. If an illegal access occurs, a Mode Mismatch interrupt is generated and reflected in the Core Interrupt register (GINTSTS.ModeMis). When the core switches from one mode to another, the registers in the new mode must be reprogrammed as they would be after a power-on reset.

25.4.1 CSR Memory Map

The CSR address map is fixed and does not depend on the core's configuration (for example, how many endpoints are implemented). Host and Device mode registers occupy different addresses. All registers are implemented in the AHB Clock domain.

Figure 25-3 shows the CSR address map.

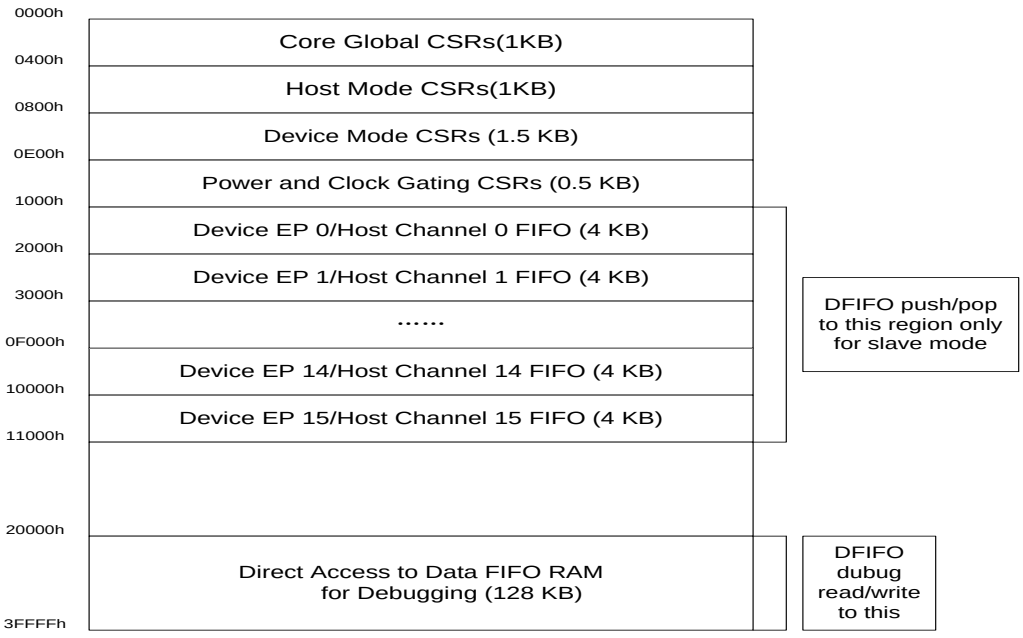


Figure 25-3 OTG CSR Memory Map

25.4.2 Register Maps

The tables in this section provide high-level summaries of each register and register group.

Register Name	Name of register types and register names ordered by offset address
Acronym	Shorthand names for registers that are mapped to the offset address. These are used extensively in the programming examples in the Programming Guide. The first letter is a prefix for the register type: G: Core Global H: Host mode D: Device mode
Offset Address	Address, in hexadecimal (h), of the first byte of each register.

Note: FIFO size and FIFO depth are used interchangeably.

25.4.3 Global CSR Map

These registers are available in both Host and Device modes.

Table 25-3 Core Global CSR Map (000h-3FFh)

Acronym	Offset Address	Register Name
GOTGCTL	000h	"Control and Status Register (GOTGCTL)"
GOTGINT	004h	"Interrupt Register (GOTGINT)"
GAHBCFG	008h	"AHB Configuration Register (GAHBCFG)"
GUSBCFG	00Ch	"USB Configuration Register (GUSBCFG)"
GRSTCTL	010h	"Reset Register (GRSTCTL)"
GINTSTS	014h	"Interrupt Register (GINTSTS)"
GINTMSK	018h	"Interrupt Mask Register (GINTMSK)"
GRXSTSR	01Ch	"Receive Status Debug Read/Status Read and Pop Registers (GRXSTSR/GRXSTSP)
GRXSTSP	020h	
GRXFSIZ	024h	"Receive FIFO Size Register (GRXFSIZ)"
GNPTXFSIZ	028h	"Non-Periodic Transmit FIFO Size Register (GNPTXFSIZ)"
GNPTXSTS	02Ch	"Non-Periodic Transmit FIFO/Queue Status Register (GNPTXSTS)"
	030h - 038h	Reserved
GUID	03Ch	"User ID Register (GUID)"
GSNPSID	040h	"Synopsys ID Register (GSNPSID)"
GHWCFG1	044h	"User HW Config1 Register (GHWCFG1)"
GHWCFG2	048h	"User HW Config2 Register (GHWCFG2)"
GHWCFG3	04Ch	"User HW Config3 Register (GHWCFG3)"
GHWCFG4	050h	"User HW Config4 Register (GHWCFG4)"
GLPMCFG	054h	"Core LPM Configuration Register (GLPMCFG)"
	0x58h	Reserved
GDFIFOCFG	05Ch	"DFIFO Software Config Register (GDFIFOCFG)"
GADPCTL	0x60h	"ADP Timer, Control and Status Register (GADPCTL)"

HPTXFSIZ	100h	"Host Periodic Transmit FIFO Size Register (HPTXFSIZ)"
DPTXFSIZn	104h-13Ch Dedicated FIFO	"Device Periodic Transmit FIFO-n Size Register (DPTXFSIZn)"
DIEPTXFn	104h-13Ch Dedicated FIFO	"Device IN Endpoint Transmit FIFO Size Register: (DIEPTXFn)"
	140h-3FFh	Reserved

25.4.4 Host Mode CSR Map

These registers must be programmed every time the core changes to Host mode.

Table 25-4 Host Mode CSR Map (400h-7FFh)

Acronym	Offset Address	Register Name
HCFG	400h	Host Configuration Register (HCFG)
HFIR	404h	Host Frame Interval Register (HFIR)
HFNUM	408h	Host Frame Number/Frame Time Remaining Register (HFNUM)
	40Ch	Reserved
HPTXSTS	410h	Host Periodic Transmit FIFO/Queue Status Register (HPTXSTS)
HAINT	414h	Host All Channels Interrupt Register (HAINT)
HAINTMSK	418h	Host All Channels Interrupt Mask Register (HAINTMSK)
HFLBAddr	41Ch	Host Frame List Base Address Register (HFLBAddr)
HPRT	440h	Host Port Control and Status Register (HPRT)
	444h-4FCh	Reserved
HCCHARn	500h - 6E0	Host Channel-n Characteristics Register (HCCHARn)
HCSPLTn	504h - 6E4	Host Channel-n Split Control Register (HCSPLTn)
HCINTn	508h - 6E8	Host Channel-n Interrupt Register (HCINTn)
HCINTMSKn	50Ch - 6EC	Host Channel-n Interrupt Mask Register (HCINTMSKn)
HCTSIZn	510h - 6F0	Host Channel-n Transfer Size Register (HCTSIZn)
	518h - 6F8	Reserved
HCDMABn	51Ch - 6FCh	Host Channel-n DMA Buffer Address Register (HCDMABn)
	6F0-7FFh	Reserved

25.4.5 Device Mode CSR Map

These registers must be programmed every time the core changes to Device mode.

Table 25-5 Device Mode CSR Map (800h-BFFh)

Acronym	Offset Address	Register Name
	800h-ACh	Device Logical IN Endpoint-Specific Registers
DCFG	800h	Device Configuration Register (DCFG)
DCTL	804h	Device Control Register (DCTL)

DSTS	808h	Device Status Register (DSTS)
	80Ch	Reserved
DIEPMSK	810h	Device IN Endpoint Common Interrupt Mask Register (DIEPMSK)
DOEPMSK	814h	Device OUT Endpoint Common Interrupt Mask Register (DOEPMSK)
DAINT	818h	Device All Endpoints Interrupt Register (DAINT)
DAINTMSK	81Ch	Device All Endpoints Interrupt Mask Register (DAINTMSK)
DTKNQR1	820h	Device IN Token Sequence Learning Queue Read Register 1 (DTKNQR1)
DTKNQR2	824h	Device IN Token Sequence Learning Queue Read Register 2 (DTKNQR2)
DTKNQR3	830h	Device IN Token Sequence Learning Queue Read Register 3 (DTKNQR3)
DTKNQR4	834h	Device IN Token Sequence Learning Queue Read Register 4 (DTKNQR4)
DVBUSDIS	828h	Device VBUS Discharge Time Register (DVBUSDIS)
DVBUSPULSE	82Ch	Device VBUS Pulsing Time Register (DVBUSPULSE)
DTHRCTL	830h	Device Threshold Control Register (DTHRCTL)
DIEPEMPMSK	834h	Device IN Endpoint FIFO Empty Interrupt Mask Register: (DIEPEMPMSK)
DEACHINT	838h	Device Each Endpoint Interrupt Register (DEACHINT)
DEACHINTMSK	83Ch	Device Each Endpoint Interrupt Register Mask (DEACHINTMSK)
DIEPEACHMSKn	840h	Device Each In Endpoint-n Interrupt Register (DIEPEACHMSKn)
DOEPEACHMSKn	880h	Device Each Out Endpoint-n Interrupt Register (DOEPEACHMSKn)
DIEPCTL0	900h	Device Control IN Endpoint 0 Control Register (DIEPCTL0)
	904h	Reserved
DIEPCTLn	920h-AE0h	Device Endpoint-n Control Register (DIEPCTLn/DOEPCTLn)
DIEPINTn		Device Endpoint-n Interrupt Register (DIEPINTn/DOEPINTn)
	90Ch	Reserved
DIEPTSIZ0	910h	Device Endpoint 0 Transfer Size Register (DIEPTSIZ0/DOEPTSIZ0)
DIEPTSIZn	910h	Device Endpoint-n Transfer Size Register (DIEPTSIZn/DOEPTSIZn)

DIEPDMAn	914h	Device Endpoint-n DMA Address Register (DIEPDMAn/DOEPDMAn)
DTXFSTSn	918h	Device IN Endpoint Transmit FIFO Status Register (DTXFSTSn)
DIEPDMAB0/DIEPDMABn	91Ch	Device Endpoint-n DMA Buffer Address Register (DIEPDMABn/DOEPDMABn)
DOEPCTL0	B00h	Device Control OUT Endpoint 0 Control Register (DOEPCTL0)
	B04h	Reserved
DOEPCTLn	B20h-CE0h	Device Endpoint-n Control Register (DIEPCTLn/DOEPCTLn)
DOEPINTn	B08h	Device Endpoint-n Interrupt Register (DIEPINTn/DOEPINTn)
	B0Ch	Reserved
DOEPTSIZ0	B10h	Device Endpoint 0 Transfer Size Register (DIEPTSIZ0/DOEPTSIZ0)
DOEPTSIZn	B30h-DB0h	Device Endpoint-n Transfer Size Register (DIEPTSIZn/DOEPTSIZn)
DOEPDMAn	B14h-CF4h	Device Endpoint-n DMA Address Register (DIEPDMAn/DOEPDMAn)
DOEPDMAB0/ DOEPDMABn	B1Ch-CFCh	Device Endpoint-n DMA Buffer Address Register (DIEPDMABn/DOEPDMABn)

25.4.6 Data FIFO (DFIFO) Access Register Map

These registers, available in both Host and Device modes, are used to read or write the FIFO space for a specific endpoint or a channel, in a given direction. If a host channel is of type IN, the FIFO can only be read on the channel. Similarly, if a host channel is of type OUT, the FIFO can only be written on the channel.

Table 25-6 Data FIFO (DFIFO) Access Register Map

FIFO Access Register Section	Address Range	Access
Device IN Endpoint 0/Host OUT Channel 0: DFIFO Write Access Device OUT Endpoint 0/Host IN Channel 0: DFIFO Read Access	1000h-1FFCh	WO/RO
Device IN Endpoint 1/Host OUT Channel 1: DFIFO Write Access Device OUT Endpoint 1/Host IN Channel 1: DFIFO Read Access	2000h-2FFCh	WO/RO
...	...	...
Device IN Endpoint 14/Host OUT Channel 14: DFIFO Write Access Device OUT Endpoint 14/Host IN Channel 14: DFIFO Read Access	F000h-FFFCh	WO/RO
Device IN Endpoint 15/Host OUT Channel 15: DFIFO Write	10000h-10FFCh	WO/RO

Access Device OUT Endpoint 15/Host IN Channel 15: DFIFO Read Access		

25.5 Register Descriptions

This section describes Core Global, Device Mode, Host Mode CSRs. Detailed register programming examples are provided in the Programming Guide.

25.5.1 Application Access to the CSRs

The Access column of each register description that follows specifies how the application and the core can access the register fields of the CSRs. The following conventions are used:

Read Only (RO)	Register field can only be read by the application. Writes to read-only fields have no effect.
Write Only (WO)	Register field can only be written by the application.
Read and Write (R_W)	Register field can be read and written by the application. The application can set this field by writing 1'b1 and can clear it by writing 1'b0.
Read, Write, and Self Clear (R_W_SC)	Register field can be read and written by the application (Read and Write), and is cleared to 1'b0 by the core (Self Clear). The conditions under which the core clears this field are explained in detail in the field's description.
Read, Write, Self Set, and Self Clear (R_W_SS_SC)	Register field can be read and written by the application (Read and Write), set to 1'b1 by the core on certain USB events (Self Set), and cleared to 1'b0 by the core (Self Clear). The conditions under which the core sets and clears this field are explained in the field's description. (Only the Port Resume bit of the Host Port Control and Status register, HPRT.PrtRes, uses this access type).
Read, Self Set, and Write Clear (R_SS_WC)	Register field can be read by the application (Read), can be set to 1'b1 by the core on a certain internal or USB or AHB event (Self Set), and can be cleared to 1'b0 by the application with a register write of 1'b1 (Write Clear). A register write of 1'b0 has no effect on this field. The conditions under which the core sets this field are explained in detail in the field's description. (For example, interrupt bits.)
Read, Write Set, and Self Clear (R_WS_SC)	Register field can be read by the application (Read), can be set to 1'b1 by the application with a register write of 1'b1 (Write Set), and is cleared to 1'b0 by the core (Self Clear). The application cannot clear this type of field, and a register write of 1'b0 to this bit has no effect on this field. The conditions under which the core clears this field are explained in detail in the field's description. (For example, reset signals)
Read, Self Set, and Self Clear or Write	Register field can be read by the application (Read), can be set to 1'b1 by the core on certain internal or USB or AHB events (Self Set), and can be

Clear (R_SS_SC_WC)	cleared to 1'b0 either by the core itself (Self Clear) or by the application with a register write of 1'b1 (Write Clear). A register write of 1'b0 to this bit has to no effect on this field. The conditions under which the core sets or clears this field are explained in the field's description. (Only the Port Enable bit of the Host Port Control and Status register, HPRT.PrtEna, and the VStatus Done bit of the PHY Vendor Control register, GPVNDCTL.VStsDone, use this access type.)

Note: Always program Reserved fields with 0s. Treat read values from Reserved fields as unknowns(Xs).

25.5.2 Overview of Commonly Used Register Bits

This section provides an overview of the commonly used registers and bits. For a complete description of all the registers, see the corresponding sections.

Table 25-7 List of Commonly Used Register Bits

Bit Number	Register/Bit Name	Description
	"Control and Status Register (GOTGCTL)"	The OTG Control and Status register controls the behavior and reflects the status of the OTG function of the core.
11	Device HNP Enabled (DevHNPEn)	The application sets this bit when it successfully receives a SetFeature.SetHNPEnable command from the connected USB host. ▪ 1'b0: HNP is not enabled in the application ▪ 1'b1: HNP is enabled in the application
10	Host Set HNP Enable (HstSetHNPEn)	The application sets this bit when it has successfully enabled HNP (using the SetFeature.SetHNPEnable command) on the connected device. ▪ 1'b0: Host Set HNP is not enabled ▪ 1'b1: Host Set HNP is enabled
	"AHB Configuration Register (GAHBCFG)"	This register can be used to configure the core after power-on or a change in mode. This register mainly contains AHB system-related configuration parameters. Do not change this register after the initial programming. The application must program this register before starting any transactions on either the AHB or the USB.
5	DMA Enable (DMAEn)	▪ 1'b0: Core operates in Slave mode ▪ 1'b1: Core operates in a DMA mode
	"USB Configuration Register (GUSBCFG)"	This register can be used to configure the core after power-on or a changing to Host mode or Device mode. It contains USB and USB-PHY related configuration parameters. The application must program this register before starting any transactions on either the

		AHB or the USB. Do not make changes to this register after the initial programming.
30	Force Device Mode (ForceDevMode)	Writing a 1 to this bit forces the core to device mode irrespective of utmiotg_iddig input pin. ▪ 1'b0: Normal Mode ▪ 1'b1: Force Device Mode A fter setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 µs is sufficient.
29	Force Host Mode (ForceHstMode)	Writing a 1 to this bit forces the core to host mode irrespective of utmiotg_iddig input pin. ▪ 1'b0: Normal Mode ▪ 1'b1: Force Host Mode After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 µs is sufficient.
9	HNP-Capable (HNPCap)	The application uses this bit to control the DWC_otg core's HNP capabilities. ▪ 1'b0: HNP capability is not enabled. ▪ 1'b1: HNP capability is enabled.
8	SRP-Capable (SRPCap)	The application uses this bit to control the DWC_otg core SRP capabilities. If the core operates as a non-SRP-capable B-device, it cannot request the connected A-device (host) to activate VBUS and start a session. ▪ 1'b0: SRP capability is not enabled. ▪ 1'b1: SRP capability is enabled.
6	USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver Select	The application uses this bit to select either a high-speed UTMI+ or ULPI PHY, or a full-speed transceiver. ▪ 1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY ▪ 1'b1: USB 1.1 full-speed serial transceiver This bit is always 0, with Read Only access.
4	ULPI or UTMI+ Select (ULPI_UTMI_Sel)	The application uses this bit to select either a UTMI+ interface or ULPI Interface. ▪ 1'b0: UTMI+ Interface ▪ 1'b1: ULPI Interface This bit is always 0, with Read Only access.
3	PHY Interface (PHYIf)	The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. 1'b0: 8 bits 1'b1: 16 bits
	"Host Configuration	This register configures the core after power-on. Do not make

	Register (HCFG)"	changes to this register after initializing the host.
23	Enable Scatter/gather DMA in Host mode (DescDMA)	When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. NOTE: This bit must be modified only once after a reset. The following combinations are available for programming: ▪ GAHBCFG.DMAEn=0, HCFG.DescDMA=0 => Slave mode ▪ GAHBCFG.DMAEn=0, HCFG.DescDMA=1 => Invalid ▪ GAHBCFG.DMAEn=1, HCFG.DescDMA=0 => Buffered DMA mode ▪ GAHBCFG.DMAEn=1, HCFG.DescDMA=1 => Scatter/Gather DMA mode In non Scatter/Gather DMA mode, this bit is reserved.
2	FS- and LS-Only Support(FSLSSupp)	The application uses this bit to control the core's enumeration speed. Using this bit, the application can make the core enumerate as a FS host, even if the connected device supports HS traffic. Do not make changes to this field after initial programming. ▪ 1'b0: HS/FS/LS, based on the maximum speed supported by the connected device ▪ 1'b1: FS/LS-only, even if the connected device can support HS
	"Host Channel-n Characteristics Register (HCCHARn)"	
19:18	Endpoint Type (EPType)	Indicates the transfer type selected. ▪ 2'b00: Control ▪ 2'b01: Isochronous ▪ 2'b10: Bulk ▪ 2'b11: Interrupt
15	Endpoint Direction (EPDir)	Indicates whether the transaction is IN or OUT. ▪ 1'b0: OUT ▪ 1'b1: IN
14:11	Endpoint Number (EPNum)	Indicates the endpoint number on the device serving as the data source or sink.
10:0	Maximum Packet Size (MPS)	Indicates the maximum packet size of the associated endpoint.
	"Host Channel-n Transfer Size Register (HCTSIZn)"	
30:29	PID (Pid)	The application programs this field with the type of PID to use for the initial transaction. The host maintains this field for the rest of the transfer. ▪ 2'b00: DATA0

		▪ 2'b01: DATA2 ▪ 2'b10: DATA1 ▪ 2'b11: MDATA (non-control)
	"Device Configuration Register (DCFG)"	This register configures the core in Device mode after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.
25:24	Periodic Scheduling Interval (PerSchIntvl)	PerSchIntvl must be programmed only for Scatter/Gather DMA mode. Description: This field specifies the amount of time the Internal DMA engine must allocate for fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25,50 or 75% of (micro)frame. ▪ When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data. ▪ When no periodic endpoints are active, then the internal DMA engine services non-periodic endpoints, ignoring this field. ▪ After the specified time within a (micro)frame, the DMA switches to fetching for non-periodic endpoints. ▪ 2'b00: 25% of (micro)frame. ▪ 2'b01: 50% of (micro)frame. ▪ 2'b10: 75% of (micro)frame. ▪ 2'b11: Reserved.
23	Enable Scatter/Gather DMA in Device mode (DescDMA)	When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. NOTE: This bit must be modified only once after a reset. The following combinations are available for programming: ▪ GAHBCFG.DMAEn=0,DCFG.DescDMA=0 => Slave mode ▪ GAHBCFG.DMAEn=0,DCFG.DescDMA=1 => Invalid ▪ GAHBCFG.DMAEn=1,DCFG.DescDMA=0 => Buffered DMA mode ▪ GAHBCFG.DMAEn=1,DCFG.DescDMA=1 => Scatter/Gather DMA mode
10:4	Device Address (DevAddr)	The application must program this field after every SetAddress control command.
1:0	Device Speed (DevSpd)	Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. See "Device Initialization" in the Programming Guide for details.

		▪ 2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) ▪ 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) ▪ 2'b10: Reserved ▪ 2'b11: Full speed (USB 1.1 transceiver clock is 48 MHz)
	"Device Endpoint-n Control Register (DIEPCTLn/DOEPCTLn)"	The application uses this register to control the behavior of each logical endpoint other than endpoint 0.
29	Set DATA1 PID (SetD1PID)	Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode.
28	Set DATA0 PID (SetD0PID)	Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode.
19:18	Endpoint Type (EPType)	Applies to IN and OUT endpoints. This is the transfer type supported by this logical endpoint. ▪ 2'b00: Control ▪ 2'b01: Isochronous ▪ 2'b10: Bulk ▪ 2'b11: Interrupt
10:00	Maximum Packet Size (MPS)	Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.

25.5.3 Global Registers

These registers are available in both Host and Device modes, and do not need to be reprogrammed when switching between these modes.

25.5.3.1 Control and Status Register (GOTGCTL)

✧ Offset: 000h

The OTG Control and Status register controls the behavior and reflects the status of the OTG function of the core.

Table 25-8 Control and Status Register: GOTGCTL

Field	Description	Mode	Reset	Access
31:26	Reserved	Host and Device		RO
27	Chirp On Enable (ChirpEn)	Device	1'b0	RW

	This bit when programmed to 1'b1 results in the core asserting chirp_on before sending an actual Chirp "K" signal on USB. This bit is present only if OTG_BC_SUPPORT = 1.If OTG_BC_SUPPORT!=1, this bit is a reserved bit.	only		
26:22	Multi Valued ID pin (MultValIdBc) Battery Charger ACA inputs in the following order: ▪ Bit 26 - rid_float. ▪ Bit 25 - rid_gnd ▪ Bit 24 - rid_a ▪ Bit 23 - rid_b ▪ Bit 22 - rid_c These bits are valid only if OTG_BC_SUPPORT=1, otherwise they are reserved.	Host and Device	Configurable. The reset value is unknown if OTG_BC_SUPPORT =1 and 0 otherwise.	RO
21	Reserved	Host and Device		RO
20	OTG Version (OTGVer) Indicates the OTG revision. ▪ 1'b0: OTG Version 1.3. In this version the core supports Data line pulsing and VBus pulsing for SRP. ▪ 1'b1: OTG Version 2.0. In this version the core supports only Data line pulsing for SRP.		1'b0	RW
19	B-Session Valid (BSesVld) Indicates the Device mode transceiver status. ▪ 1'b0: B-session is not valid. ▪ 1'b1: B-session is valid. In OTG mode, you can use this bit to determine if the device is connected or disconnected. **Note:** If you do not enable OTG features (such as SRP and HNP), the read reset value will be 1.The vbus assigns the values internally for non- SRP or non-HNP configurations. In case of OTG_MODE=0, the reset value of this bit is 1'b0.	Device only	Configurable	RO
18	A-Session Valid (ASesVld) Indicates the Host mode transceiver status. ▪ 1'b0: A-session is not valid ▪ 1'b1: A-session is valid **Note:** If you do not enable OTG features (such as SRP and HNP), the read reset value will be 1.The vbus assigns the values internally for non- SRP or non-HNP configurations.	Host only	Configurable	RO

	In device mode, this bit is reserved.			
17	Long/Short Debounce Time (DbncTime) Indicates the debounce time of a detected connection. ▪ 1'b0: Long debounce time, used for physical connections (100 ms + 2.5 μs) ▪ 1'b1: Short debounce time, used for soft connections (2.5 μs)	Host only	1'b0	RO
16	Connector ID Status (ConIDSts) Indicates the connector ID status on a connect event. ▪ 1'b0: The DWC_otg core is in A-Device mode ▪ 1'b1: The DWC_otg core is in B-Device mode	Host and Device	1'b1	RO
15:12	Reserved	Host and Device		RO
11	Device HNP Enabled (DevHNPEn) The application sets this bit when it successfully receives a SetFeature.SetHNPEnable command from the connected USB host. ▪ 1'b0: HNP is not enabled in the application ▪ 1'b1: HNP is enabled in the application	Device Only OTG configurations	1'b0	R_W
10	Host Set HNP Enable (HstSetHNPEn) The application sets this bit when it has successfully enabled HNP (using the SetFeature.SetHNPEnable command) on the connected device. ▪ 1'b0: Host Set HNP is not enabled ▪ 1'b1: Host Set HNP is enabled	Device Only OTG configurations	1'b0	R_W
9	HNP Request (HNPReq) The application sets this bit to initiate an HNP request to the connected USB host. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. ▪ 1'b0: No HNP request ▪ 1'b1: HNP request	Device Only OTG configurations	1'b0	R_W
8	Host Negotiation Success (HstNegScs) The core sets this bit when host negotiation is successful. The core clears this bit when the HNP Request (HNPReq) bit in this register is set. ▪ 1'b0: Host negotiation failure ▪ 1'b1: Host negotiation success	Device only	1'b0	RO
7	B-Peripheral Session Valid OverrideValue (BvalidOvVal) This bit is used to set the Override value for Bvalid signal when GOTGCTL.BvalidOvEn is set.	Device only	1'b0	R_W

	▪ 1'b0: Bvalid value is 1'b0 when GOTGCTL.BvalidOvEn = 1. ▪ 1'b1: Bvalid value is 1'b1 when GOTGCTL.BvalidOvEn = 1.			
6	B-Peripheral Session Valid Override Enable (BvalidOvEn) This bit is used to enable/disable the software to override the Bvalid signal using the GOTGCTL.BvalidOvVal. ▪ 1'b1: Internally Bvalid received from the PHY is overridden with GOTGCTL.BvalidOvVal. ▪ 1'b0: Override is disabled and Bvalid signal from the respective PHY selected is used internally by the core.	Device only	1'b0	R_W
5	A-Peripheral Session Valid OverrideValue (AvalidOvVal) This bit is used to set the Override value for Avalid signal when GOTGCTL.AvalidOvEn is set. ▪ 1'b0: Avalid value is 1'b0 when GOTGCTL.AvalidOvEn = 1. ▪ 1'b1: Avalid value is 1'b1 when GOTGCTL.AvalidOvEn = 1.	Host only	1'b0	R_W
4	A-Peripheral Session Valid Override Enable (AvalidOvEn) This bit is used to enable/disable the software to override the Avalid signal using the GOTGCTL.AvalidOvVal. ▪ 1'b1: Internally Avalid received from the PHY is overridden with GOTGCTL.AvalidOvVal. ▪ 1'b0: Override is disabled and Avalid signal from the respective PHY is used internally by the core.	Host only	1'b0	R_W
3	VBUS Valid OverrideValue (VbvalidOvVal) This bit is used to set the Override value for vbus valid signal when GOTGCTL.VbusvalidOvEn is set. ▪ 1'b0: vbusvalid value is 1'b0 when GOTGCTL.VbvalidOvEn = 1. ▪ 1'b1: vbusvalid value is 1'b1 when GOTGCTL.VbvalidOvEn = 1.	Host only	1'b0	R_W
2	VBUS Valid Override Enable (VbvalidOvEn) This bit is used to enable/disable the software to override the vbus-valid signal using the GOTGCTL.vbvalidOvVal. ▪ 1'b1: The vbus-valid signal received from the PHY is overridden with GOTGCTL.vbvalidOvVal. ▪ 1'b0: Override is disabled and avalid signal from the respective PHY is used internally by the core.	Host only	1'b0	R_W
1	Session Request (SesReq) The application sets this bit to initiate a session request on the USB. The application can clear this bit by writing a	Device only	1'b0	R_W

	0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. If you use the USB 1.1 Full-Speed Serial Transceiver interface to initiate the session request, the application must wait until the VBUS discharges to 0.2 V, after the B-Session Valid bit in this register (GOTGCTL.BSesVld) is cleared. This discharge time varies between different PHYs and can be obtained from the PHY vendor. ▪ 1'b0: No session request ▪ 1'b1: Session request			
0	Session Request Success (SesReqScs) The core sets this bit when a session request initiation is successful. ▪ 1'b0: Session request failure ▪ 1'b1: Session request success	Device only	1'b0	RO

25.5.3.2 Interrupt Register (GOTGINT)

✧ Offset: 004h

The application reads this register whenever there is an OTG interrupt and clears the bits in this register to clear the OTG interrupt.

Table 25-9 Interrupt Register: GOTGINT

Field	Description	Mode	Reset	Access
31:20	Reserved	Host and Device		RO
20	Multi-Valued input changed This bit when set indicates that there is a change in the value of at least one ACA pin value. This bit is present only if OTG_BC_SUPPORT = 1, otherwise it is reserved.		1'b0	R_SS_WC
19	Debounce Done (DbnceDone) The core sets this bit when the debounce is completed after the device connect. The application can start driving USB reset after seeing this interrupt. This bit is only valid when the HNP Capable or SRP Capable bit is set in the Core USB Configuration register (GUSBCFG.HNPCap or GUSBCFG.SRPCap, respectively).	Host Only	1'b0	R_SS_WC
18	A-Device Timeout Change (ADevTOUTChg) The core sets this bit to indicate that the A-device has	Host and Device	1'b0	R_SS_WC

	timed out while waiting for the B-device to connect.			
17	Host Negotiation Detected (HstNegDet) The core sets this bit when it detects a host negotiation request on the USB.	Host and Device	1'b0	R_SS_WC
16：10	Reserved	Host and Device		RO
9	Host Negotiation Success Status Change (HstNegSucStsChng) The core sets this bit on the success or failure of a USB host negotiation request. The application must read the Host Negotiation Success bit of the OTG Control and Status register (GOTGCTL.HstNegScs) to check for success or failure.	Host and Device	1'b0	R_SS_WC
8	Session Request Success Status Change (SesReqSucStsChng) The core sets this bit on the success or failure of a session request. The application must read the Session Request Success bit in the OTG Control and Status register (GOTGCTL.SesReqScs) to check for success or failure.	Host and Device	1'b0	R_SS_WC
7:3	Reserved	Host and Device		RO
2	Session End Detected (SesEndDet) The core sets this bit when the utmisrp_bvalid signal is deasserted.	Device only	1'b0	R_SS_WC
1:0	Reserved	Host and Device		RO

25.5.3.3 AHB Configuration Register (GAHBCFG)

✧ Offset: 008h

This register can be used to configure the core after power-on or a change in mode. This register mainly contains AHB system-related configuration parameters. Do not change this register after the initial programming. The application must program this register before starting any transactions on either the AHB or the USB.

Table 25-10 AHB Configuration Register: GAHBCFG

Field	Description	Mode	Reset	Access
31:25	Reserved	Host and Device		RO
24	Inverse Descriptor Endianness(InvDescEndianness) 1'b0: Descriptor endianness is similar to the AHB Master	Host and Device	1’b0	R_W

	endianness 1'b1: If the AHB Master endianness is Big Endian, the Descriptor Endianness is Little Endian. If the AHB Master endianness is Little Endian, the Descriptor Endianness is Big Endian.			
23	AHB Single Support (AHBSingle) This bit when programmed supports Single transfers for the remaining data in a transfer when the DWC_otg core is operating in DMA mode. ▪ 1'b0: This is the default mode. When this bit is set to 1'b0, the remaining data in the transfer is sent using INCR burst size. ▪ 1'b1: When set to 1'b1, the remaining data in a transfer is sent using Single burst size. Note: if this feature is enabled, the AHB RETRY and SPLIT transfers still have INCR burst type. Enable this feature when the AHB Slave connected to the DWC_otg core does not support INCR burst (and when Split, and Retry transactions are not being used in the bus).	Host and Device	1'b0	R_W
22	Notify All Dma Write Transactions (NotiAllDmaWrit) This bit is programmed to enable the System DMA Done functionality for all the DMA write Transactions corresponding to the Channel/Endpoint. This bit is valid only when GAHBCFG.RemMemSupp is set to 1. ▪ GAHBCFG.NotiAllDmaWrit = 1 – HS OTG core asserts int_dma_req for all the DMA write transactions on the AHB interface along with int_dma_done, chep_last_transact and chep_number signal informations. The core waits for sys_dma_done signal for all the DMA write transactions in order to complete the transfer of a particular Channel/Endpoint. ▪ GAHBCFG.NotiAllDmaWrit = 0 – HS OTG core asserts int_dma_req signal only for the last transaction of DMA write transfer corresponding to a particular Channel/Endpoint. Similarly, the core waits for sys_dma_done signal only for that transaction of DMA write to complete the transfer of a particular Channel/Endpoint.	Host and Device	1'b0	R_W
21	Remote Memory Support (RemMemSupp) This bit is programmed to enable the functionality to wait for the system DMA Done Signal for the DMA Write	Host and Device	1'b0	R_W

	Transfers. ▪ GAHBCFG.RemMemSupp=1 – The int_dma_req output signal is asserted when HS OTG DMA starts write transfer to the external memory. When the core is done with the Transfers it asserts int_dma_done signal to flag the completion of DMA writes from HS OTG. The core then waits for sys_dma_done signal from the system to proceed further and complete the Data Transfer corresponding to a particular Channel/Endpoint. ▪ GAHBCFG.RemMemSupp=0 – The int_dma_req and int_dma_done signals are not asserted and the core proceeds with the assertion of the XferComp interrupt as soon as the DMA write transfer is done at the HS OTG Core Boundary and it doesn't wait for the sys_dma_done signal to complete the DATA transfers.			
20:9	Reserved	Host and Device		RO
8	Periodic TxFIFO Empty Level (PTxFEmpLvl) Indicates when the Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.PTxFEmp) is triggered. This bit is used only in Slave mode. ▪ 1'b0: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is half empty ▪ 1'b1: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is completely empty	Host only	1'b0	R_W
7	Non-Periodic TxFIFO Empty Level (NPTxFEmpLvl) This bit is used only in Slave mode. In host mode and with Shared FIFO with device mode, this bit indicates when the Non-Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.NPTxFEmp) is triggered. With dedicated FIFO in device mode, this bit indicates when IN endpoint Transmit FIFO empty interrupt (DIEPINTn.TxFEmp) is triggered. Host mode and with Shared FIFO with device mode: ▪ 1'b0: GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is half empty ▪ 1'b1: GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is completely empty	Host and Device	1'b0	R_W

	Dedicated FIFO in device mode: ▪ 1'b0: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is half empty ▪ 1'b1: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is completely empty			
6	Reserved	Host and Device		RO
5	DMA Enable (DMAEn) ▪ 1'b0: Core operates in Slave mode ▪ 1'b1: Core operates in a DMA mode	Host and Device	1'b0	R_W
4:1	Burst Length/Type (HBstLen) This field is used in both External and Internal DMA modes. In External DMA mode, these bits appear on dma_burst[3:0] ports, which can be used by an external wrapper to interface the External DMA Controller interface to Synopsys DW_ahb_dmac or ARM PrimeCell. External DMA Mode — defines the DMA burst length in terms of 32-bit words: ▪ 4'b0000: 1 word ▪ 4'b0001: 4 words ▪ 4'b0010: 8 words ▪ 4'b0011: 16 words ▪ 4'b0100: 32 words ▪ 4'b0101: 64 words ▪ 4'b0110: 128 words ▪ 4'b0111: 256 words ▪ Others: Reserved Internal DMA Mode — AHB Master burst type: ▪ 4'b0000 Single ▪ 4'b0001 INCR ▪ 4'b0011 INCR4 ▪ 4'b0101 INCR8 ▪ 4'b0111 INCR16 ▪ Others: Reserved Slave Mode: Only single burst is supported.	Host and Device	4'b0	R_W
0	Global Interrupt Mask (GlblIntrMsk) The application uses this bit to mask or unmask the interrupt line assertion to itself. Irrespective of this bit's setting, the interrupt status registers are updated by the core. ▪ 1'b0: Mask the interrupt assertion to the application. ▪ 1'b1: Unmask the interrupt assertion to the application.	Host and Device	1'b0	R_W

25.5.3.4 USB Configuration Register (GUSBCFG)

✧ Offset: 00Ch

This register can be used to configure the core after power-on or a changing to Host mode or Device mode.

It contains USB and USB-PHY related configuration parameters. The application must program this register before starting any transactions on either the AHB or the USB. Do not make changes to this register after the initial programming.

Table 25-11 USB Configuration Register: GUSBCFG

Field	Description	Mode	Reset	Access
31	Corrupt Tx packet This bit is for debug purposes only. Never set this bit to 1.	Host and Device	1'b0	WO
30	Force Device Mode (ForceDevMode) Writing a 1 to this bit forces the core to device mode irrespective of utmiotg_iddig input pin. ▪ 1'b0: Normal Mode ▪ 1'b1: Force Device Mode After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 µs is sufficient.	Host and Device	1'b0	R_W
29	Force Host Mode (ForceHstMode) Writing a 1 to this bit forces the core to host mode irrespective of utmiotg_iddig input pin. ▪ 1'b0: Normal Mode ▪ 1'b1: Force Host Mode After setting the force bit, the application must wait at least 25 ms before the change to take effect. When the simulation is in scale down mode, waiting for 500 µs is sufficient.	Host and Device	1'b0	R_W
28	Tx End Delay (TxEndDelay) Writing a 1 to this bit enables the TxEndDelay timers in the core as per the section 4.1.5 on Opmode of the USB 2.0 Transceiver Macrocell Interface (UTMI) version 1.05. ▪ 1'b0: Normal mode ▪ 1'b1: Introduce Tx end delay timers	Device Only	1'b0	R_W
27	IC_USB TrafficPullRemove Control (IC_USBTrafCtl) When this bit is set, pullup/pulldown resistors are detached from the USB during traffic signaling, per section 6.3.4 of the IC_USB	Device	1'h0	R_W

	specification. This bit is invalid.			
26	IC_USB-Capable (IC_USBCap) The application uses this bit to control the DWC_otg core's IC_USB capabilities. ▪ 1'b0: IC_USB PHY Interface is not selected. ▪ 1'b1: IC_USB PHY Interface is selected. This bit is set to 1'b0 and the bit is read only.	Host and Device	Read description	RO/R_W
25	ULPI Interface Protect Disable Controls circuitry built into the PHY for protecting the ULPI interface when the link tri-states STP and data. Any pull-ups or pull-downs employed by this feature can be disabled. ▪ 1'b0: Enables the interface protect circuit ▪ 1'b1: Disables the interface protect circuit This bit is reserved and read-only.	Host only	1'b0	R_W/RO
24	Indicator Pass Through Controls wether the Complement Output is qualified with the Internal Vbus Valid comparator before being used in the Vbus State in the RX CMD. Please refer to the ULPI Specification for more detail. ▪ 1'b0: Complement Output signal is qualified with the Internal VbusValid comparator. ▪ 1'b1: Complement Output signal is not qualified with the Internal VbusValid comparator. This bit is reserved and read-only.	Host Only	1'b0	R_W/RO
23	Indicator Complement Controls the PHY to invert the ExternalVbusIndicator input signal, generating the Complement Output. Please refer to the ULPI Specification for more detail ▪ 1'b0: PHY does not invert ExternalVbusIndicator signal ▪ 1'b1: PHY does invert ExternalVbusIndicator signal This bit is reserved and read-only.	Host Only	1'b0	R_W/RO
22	TermSel DLine Pulsing Selection (TermSelDLPulse) This bit selects utmi_termselect to drive data line pulse during SRP. ▪ 1'b0: Data line pulsing using utmi_txvalid (default). ▪ 1'b1: Data line pulsing using utmi_termsel.	Device Only	1'b0	R_W
21	ULPI External VBUS Indicator (ULPIExtVbusIndicator)	Host	1'b0	R_W

	This bit indicates to the ULPI PHY to use an external VBUS over-current indicator. ▪ 1'b0: PHY uses internal VBUS valid comparator. ▪ 1'b1: PHY uses external VBUS valid comparator. This bit is invalid.	Only		
20	ULPI External VBUS Drive (ULPIExtVbusDrv) This bit selects between internal or external supply to drive 5V on VBUS, in ULPI PHY. ▪ 1'b0: PHY drives VBUS using internal charge pump (default). ▪ 1'b1: PHY drives VBUS using external supply. This bit is invalid.	Host Only	1'b0	R_W
19	ULPI Clock SuspendM (ULPIClkSusM) This bit sets the ClockSuspendM bit in the Interface Control register on ULPI PHY. 1'b0: PHY powers down internal clock during suspend. 1'b1: PHY does not power down internal clock. This bit is invalid.	Host and Device	1'b0	R_W
18	ULPI Auto Resume (ULPIAutoRes) This bit sets the AutoResume bit in the Interface Control register on the ULPIPHY. ▪ 1'b0: PHY does not use AutoResume feature. ▪ 1'b1: PHY uses AutoResume feature. This bit is invalid.	Host and Device	1'b0	R_W
17	ULPI FS/LS Select (ULPIFsLs) The application uses this bit to select the FS/LS serial interface for the ULPI PHY. This bit is valid only when the FS serial transceiver is selected on the ULPI PHY. ▪ 1'b0: ULPI interface ▪ 1'b1: ULPI FS/LS serial interface	Host and Device	1'b0	R_W
16	UTMIFS or I^2C Interface Select (OtgI2CSel) The application uses this bit to select the I^2C interface. ▪ 1'b0: UTMI USB 1.1 Full-Speed interface for OTG signals ▪ 1'b1: I 2C interface for OTG signals This bit is writable only if I ^2C and UTMIFS were specified for Enable I2C Interface	Host and Device	1'b0	RO/R_W
15	PHY Low-Power Clock Select (PhyLPwrClkSel) Selects either 480-MHz or 48-MHz (low-power) PHY mode. In FS and LS modes, the PHY can usually operate on a 48-MHz clock to save power. ▪ 1'b0: 480-MHz Internal PLL clock ▪ 1'b1: 48-MHz External Clock	Host and Device	1'b0	R_W

	In 480 MHz mode, the UTMI interface operates at either 60 or 30-MHz, depending upon whether 8- or 16-bit data width is selected. In 48-MHz mode, the UTMI interface operates at 48 MHz in FS and LS modes. This bit drives the utmi_fsls_low_power core output signal, and is valid only for UTMI+ PHYs.			
14	Reserved	Host and Device		RO
13:10	USB Turnaround Time (USBTrdTim) Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (SPRAM). This must be programmed to ▪ 4'h5: When the MAC interface is 16-bit UTMI+. ▪ 4'h9: When the MAC interface is 8-bit UTMI+. **Note:** The values above are calculated for the minimum AHB frequency of 30 MHz. USB turnaround time is critical for certification where long cables and 5-Hubs are used, so if you need the AHB to run at less than 30 MHz, and if USB turnaround time is not critical, these bits can be programmed to a larger value.	Device Only	4'h5	R_W
9	HNP-Capable (HNPCap) The application uses this bit to control the DWC_otg core's HNP capabilities. ▪ 1'b0: HNP capability is not enabled. ▪ 1'b1: HNP capability is enabled. If HNP functionality is disabled by the software, the OTG signals on the PHY domain must be tied to the appropriate values.	Host and Device	1'h0	RO/R_W
8	SRP-Capable (SRPCap) The application uses this bit to control the DWC_otg core SRP capabilities. If the core operates as a non-SRP-capable B-device, it cannot request the connected A-device (host) to activate VBUS and start a session. ▪ 1'b0: SRP capability is not enabled. ▪ 1'b1: SRP capability is enabled. If SRP functionality is disabled by the software, the OTG signals on the PHY domain must be tied to the appropriate values.	Host and Device	1'h0	R_W
7	ULPI DDR Select (DDRSel) The application uses this bit to select a Single Data Rate	Host and Device	1'h0	R_W

	(SDR) or Double Data Rate (DDR) or ULPI interface. ▪ 1'b0: Single Data Rate ULPI Interface, with 8-bit-wide data bus ▪ 1'b1: Double Data Rate ULPI Interface, with 4-bit-wide data bus This bit is invalid.			
6	USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver Select (PHYSel) The application uses this bit to select either a high-speed UTMI+ or ULPI PHY, or a full-speed transceiver. ▪ 1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY ▪ 1'b1: USB 1.1 full-speed serial transceiver This bit is always 0, with Read Only access.	Host and Device	1'h0	RO/R_W
5	Full-Speed Serial Interface Select (FSIntf) The application uses this bit to select either a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface. ▪ 1'b0: 6-pin unidirectional full-speed serial interface ▪ 1'b1: 3-pin bidirectional full-speed serial interface This bit is always 0, with Read Only access.	Host and Device	1'h0	RO/R_W
4	ULPI or UTMI+ Select (ULPI_UTMI_Sel) The application uses this bit to select either a UTMI+ interface or ULPI Interface. ▪ 1'b0: UTMI+ Interface ▪ 1'b1: ULPI Interface This bit takes effect only if GUSBCFG.PHYSel = 1'b0.	Host and Device	1'h0	RO/R_W
3	PHY Interface (PHYIf) The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. When a ULPI PHY is chosen, this must be set to 8-bit mode. ▪ 1'b0: 8 bits ▪ 1'b1: 16 bits	Host and Device	Configurable	RO/R_W
2:0	HS/FS Timeout Calibration (TOutCal) The number of PHY clocks that the application programs in this field is added to the high-speed/full-speed interpacket timeout duration in the core to account for any additional delays introduced by the PHY. This can be required, because the delay introduced by the PHY in generating the line state condition can vary from one PHY to another. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for full-speed operation is 16 to 18	Host and Device	3'h0	R_W

	(inclusive) bit times. The application must program this field based on the speed of enumeration. The number of bit times added per PHY clock are: High-speed operation: ▪ One 30-MHz PHY clock = 16 bit times ▪ One 60-MHz PHY clock = 8 bit times Full-speed operation: ▪ One 30-MHz PHY clock = 0.4 bit times ▪ One 60-MHz PHY clock = 0.2 bit times ▪ One 48-MHz PHY clock = 0.25 bit times Using the HS as an example, if you set ToutCal to '001' you add one 30MHz PHY clock or 16 bit times. If you set ToutCal to '010' you add two 30MHz PHY clocks or 32 bit times, and so on. The 3 bits allow you to add up to 7 PHY clocks, and the number of bit times depend on the speed, and the PHY clock you are using.			

25.5.3.5 Reset Register (GRSTCTL)

✧ Offset: 010h

The application uses this register to reset various hardware features inside the core.

Table 25-12 Reset Register: GRSTCTL

Field	Description	Mode	Reset	Access
31	AHB Master Idle (AHBIdle) Indicates that the AHB Master State Machine is in the IDLE condition.	Host and Device	1'b1	RO
30	DMA Request Signal (DMAReq) Indicates that the DMA request is in progress. Used for debug.	Host and Device	1'b1	RO
29:11	Reserved	Host and Device		RO
10:6	TxFIFO Number (TxFNum) This is the FIFO number that must be flushed using the TxFIFO Flush bit. This field must not be changed until the core clears the TxFIFO Flush bit. ▪ 5'h0: – Non-periodic TxFIFO flush in Host mode – Non-periodic TxFIFO flush in device mode when in shared FIFO operation – Tx FIFO 0 flush in device mode when in dedicated FIFO mode	Host and Device	5'b0	R_W

	▪ 5'h1: – Periodic TxFIFO flush in Host mode – Periodic TxFIFO 1 flush in Device mode when in shared FIFO operation – TXFIFO 1 flush in device mode when in dedicated FIFO mode ▪ 5'h2: – Periodic TxFIFO 2 flush in Device mode when in shared FIFO operation – TXFIFO 2 flush in device mode when in dedicated FIFO mode ... ▪ 5'hF: – Periodic TxFIFO 15 flush in Device mode when in shared FIFO operation – TXFIFO 15 flush in device mode when in dedicated FIFO mode ▪ 5'h10: – Flush all the transmit FIFOs in device or host mode.			
5	TxFIFO Flush (TxFFlsh) This bit selectively flushes a single or all transmit FIFOs, but cannot do so if the core is in the midst of a transaction. The application must write this bit only after checking that the core is neither writing to the TxFIFO nor reading from the TxFIFO. Verify using these registers: ▪ Read—NAK Effective Interrupt ensures the core is not reading from the FIFO. ▪ Write—GRSTCTL.AHBIdle ensures the core is not writing anything to the FIFO. Flushing is normally recommended when FIFOs are re-configured or when switching between Shared FIFO and Dedicated Transmit FIFO operation. FIFO flushing is also recommended during device endpoint disable. The application must wait until the core clears this bit before performing any operations. This bit takes eight clocks to clear, using the slower clock of phy_clk or hclk.	Host and Device	1'b0	R_WS_SC
4	RxFIFO Flush (RxFFlsh) The application can flush the entire RxFIFO using this bit, but must first ensure that the core is not in the middle of a transaction. The application must only write to this bit after checking	Host and Device	1'b0	R_WS_SC

	that the core is neither reading from the RxFIFO nor writing to the RxFIFO. The application must wait until the bit is cleared before performing any other operations. This bit requires 8 clocks (slowest of PHY or AHB clock) to clear.			
3	IN Token Sequence Learning Queue Flush (INTknQFlsh) This bit is valid only if OTG_EN_DED_TX_FIFO = 0. The application writes this bit to flush the IN Token Sequence Learning Queue.	Device Only	1'b0	R_WS_SC
2	Host Frame Counter Reset (FrmCntrRst) The application writes this bit to reset the (micro)frame number counter inside the core. When the (micro)frame counter is reset, the subsequent SOF sent out by the core has a (micro)frame number of 0.	Host Only	1'b0	R_WS_SC
1	Reserved	Host and Device		RO
0	Core Soft Reset (CSftRst) Resets the hclk and phy_clock domains as follows: ▪ Clears the interrupts and all the CSR registers except the following register bits: – PCGCCTL.RstPdwnModule – PCGCCTL.GateHclk – PCGCCTL.PwrClmp – GUSBCFG.DDRSel – GUSBCFG.PHYSel – GUSBCFG.FSIntf – GUSBCFG.ULPI_UTMI_Sel – GUSBCFG.PHYIf – HCFG.FSLSPclkSel – DCFG.DevSpd – DCTL.SftDiscon – GGPIO – GUSBCFG.TxEndDelay – GUSBCFG.TermSelDLPulse – GUSBCFG.ULPIClkSusM – GUSBCFG.ULPIAutoRes – GUSBCFG.ULPIFsLs – GPWRDN – GADPCTL ▪ All module state machines (except the AHB Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed. ▪ Any transactions on the AHB Master are terminated	Host and Device	1'b0	R_WS_SC

	as soon as possible, after gracefully completing the last data phase of an AHB transfer. Any transactions on the USB are terminated immediately. ▪ When Hibernation or ADP feature is enabled, the PMU module is not reset by the Core Soft Reset. The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which can take several clocks, depending on the current state of the core. Once this bit is cleared software must wait at least 3 PHY clocks before doing any access to the PHY domain (synchronization delay). Software must also must check that bit 31 of this register is 1 (AHB Master is IDLE) before starting any operation. Typically software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain has to be reset for proper operation.			

25.5.3.6 Interrupt Register (GINTSTS)

✧ Offset: 014h

This register interrupts the application for system-level events in the current mode (Device mode or Host mode).

Some of the bits in this register are valid only in Host mode, while others are valid in Device mode only. This register also indicates the current mode. to clear the interrupt status bits of type R_SS_WC, the application must write 1'b1 into the bit.

The FIFO status interrupts are read only; once software reads from or writes to the FIFO while servicing these interrupts, FIFO interrupt conditions are cleared automatically.

The application must clear the GINTSTS register at initialization before unmasking the interrupt bit to avoid any interrupts generated prior to initialization.

Table 25-13 Interrupt Register: GINTSTS

Field	Description	Mode	Reset	Access
31	Resume/Remote Wakeup Detected Interrupt (WkUpInt) Wakeup Interrupt during Suspend(L2) or LPM(L1) state. ▪ During Suspend(L2):	Host and Device	1'b0	R_SS_WC

	– Device Mode - This interrupt is asserted only when Host Initiated Resume is detected on USB. – Host Mode - This interrupt is asserted only when Device Initiated Remote Wakeup is detected on USB. ▪ During LPM(L1): – Device Mode - This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB. – Host Mode - This interrupt is asserted for either Host Initiated Resume or Device Initiated Remote Wakeup on USB.			
30	Session Request/New Session Detected Interrupt (SessReqInt) In Host mode, this interrupt is asserted when a session request is detected from the device. In Host mode, this interrupt is asserted when a session request is detected from the device. In Device mode, this interrupt is asserted when the utmisrp_bvalid signal goes high. For more information on how to use this interrupt, see "FIFO RAM Allocation" in the Programming Guide.	Host and Device	1'b0	R_SS_WC
29	Disconnect Detected Interrupt (DisconnInt) This interrupt is asserted when a device disconnect is detected.	Host Only	1'b0	R_SS_WC
28	Connector ID Status Change (ConIDStsChng) This interrupt is asserted when there is a change in connector ID status.	Host and Device	1'b0	R_SS_WC
27	LPM Transaction Received Interrupt (LPM_Int) ▪ Device Mode - This interrupt is asserted when the device receives an LPM transaction and responds with a non-ERRORed response. ▪ Host Mode - This interrupt is asserted when the device responds to an LPM transaction with a non-ERRORed response or when the host core has completed LPM transactions for the programmed number of times (GLPMCFG.RetryCnt). This field is valid only if the Core LPM Configuration register's LPM- Capable (LPMCap) field is set to 1 and the User HW Config3 register's OTG_ENABLE_LPM bit is set to 1.	Host and Device	1'b0	R_SS_WC
26	Periodic TxFIFO Empty (PTxFEmp)	Host	1'b1	RO

	This interrupt is asserted when the Periodic Transmit FIFO is either half or completely empty and there is space for at least one entry to be written in the Periodic Request Queue. The half or completely empty status is determined by the Periodic TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.PTxFEmpLvl).	Only		
25	Host Channels Interrupt (HChInt) The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred, and then read the corresponding Host Channel-n Interrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this bit.	Host Only	1'b0	RO
24	Host Port Interrupt (PrtInt) The core sets this bit to indicate a change in port status of one of the DWC_otg core ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host Port Control and Status register to clear this bit.	Host Only	1'b0	RO
23	Reset Detected Interrupt (ResetDet) The core sets this status bit in device mode when reset is detected on the USB in L2 Suspend state. This bit is valid only when the core is Device Mode and is operating in Partial Power-Down, or Clock Gating modes of Suspend. This bit is not valid when device is in Hibernation mode of Suspend.	Device Only	1'b0	R_SS_WC
22	Data Fetch Suspended (FetSusp) This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data for IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application for an endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application: ▪ Sets a global non-periodic IN NAK handshake	Device Only	1'b0	R_SS_WC

	▪ Disables In endpoints ▪ Flushes the FIFO ▪ Determines the token sequence from the IN Token Sequence Learning Queue ▪ Re-enables the endpoints ▪ Clears the global non-periodic IN NAK handshake If the global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token is received: the core generates an "IN token received when FIFO empty" interrupt. The OTG then sends the host a NAK response. To avoid this scenario, the application can check the GINTSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a global NAK handshake. Alternatively, the application can mask the "IN token received when FIFO empty" interrupt when clearing a global IN NAK handshake.			
21	Incomplete Periodic Transfer (incomplP) In Host mode, the core sets this interrupt bit when there are incomplete periodic transactions still pending which are scheduled for the current microframe. Incomplete Isochronous OUT Transfer (incompISOOUT) The Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.	Device only Device only		
20	Incomplete Isochronous IN Transfer (incompISOIN) The core sets this interrupt to indicate that there is at least one isochronousIN endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register. Note: This interrupt is not asserted in Scatter/Gather DMA mode.	Device only	1'b0	R_SS_WC
19	OUT Endpoints Interrupt (OEPInt) The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding Device OUT Endpoint-*n* Interrupt (DOEPINTn) register to determine the exact cause of the	Device only	1'b0	RO

	interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit.			
18	IN Endpoints Interrupt (IEPInt) The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on which the interrupt occurred, and then read the corresponding Device IN Endpoint-*n* Interrupt (DIEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINTn register to clear this bit.	Device only	1'b0	RO
17	Endpoint Mismatch Interrupt (EPMis) **Note**: This interrupt is valid only in shared FIFO operation. ndicates that an IN token has been received for a non-periodic endpoint, but the data for another endpoint is present in the top of the Non-periodic Transmit FIFO and the IN endpoint mismatch count programmed by the application has expired.	Device only	1'b0	R_SS_WC
16	Restore Done Interrupt (RstrDoneInt) The core sets this bit to indicate that the restore command after Hibernation was completed by the core. The core continues from Suspended state into the mode dictated by PCGCCTL.RestoreMode field. This bit is valid only when Hibernation feature is enabled (OTG_EN_PWROPT=2)	Host and Device mode	1'b0	R_SS_WC
15	End of Periodic Frame Interrupt (EOPF) Indicates that the period specified in the Periodic Frame Interval field of the Device Configuration register (DCFG.PerFrInt) has been reached in the current microframe.	Device only	1'b0	R_SS_WC
14	Isochronous OUT Packet Dropped Interrupt (ISOOutDrop) The core sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO does not have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint.	Device only	1'b0	R_SS_WC
13	Enumeration Done (EnumDone) The core sets this bit to indicate that speed enumeration is complete. The application must read the Device Status (DSTS) register to obtain the enumerated speed.	Device only	1'b0	R_SS_WC
12	USB Reset (USBRst) The core sets this bit to indicate that a reset is detected on	Device only	1'b0	R_SS_WC

	the USB.			
11	USB Suspend (USBSusp) The core sets this bit to indicate that a suspend was detected on the USB. The core enters the Suspended state when there is no activity on the utmi_linestate signal for an extended period of time.	Device only	1'b0	R_SS_WC
10	Early Suspend (ErlySusp) The core sets this bit to indicate that an Idle state has been detected on the USB for 3 ms.	Device only	1'b0	R_SS_WC
9	I^2C Interrupt (I2CINT) The core sets this interrupt when I^2C access is completed on the I^2C interface. This field is used only if the I^2C interface was enabled. Otherwise, reads return 0.	Host and Device	1'b0	R_SS_WC
8	ULPI Carkit Interrupt (ULPICKINT) This field is used only if the Carkit interface was enabled. Otherwise, reads return 0.	Host and Device	1'b0	R_SS_WC
7	Global OUT NAK Effective (GOUTNakEff) Indicates that the Set Global OUT NAK bit in the Device Control register (DCTL.SGOUTNak), set by the application, has taken effect in the core. This bit can be cleared by writing the Clear Global OUT NAK bit in the Device Control register (DCTL.CGOUTNak).	Device only	1'b0	RO
6	Global IN Non-Periodic NAK Effective (GINNakEff) Indicates that the Set Global Non-periodic IN NAK bit in the Device Control register (DCTL.SGNPInNak), set by the application, has taken effect in the core. That is, the core has sampled the Global IN NAK bit set by the application. This bit can be cleared by clearing the Clear Global Non-periodic IN NAK bit in the Device Control register (DCTL.CGNPInNak). This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit.	Device only	1'b0	RO
5	Non-Periodic TxFIFO Empty (NPTxFEmp) This interrupt is valid only when OTG_EN_DED_TX_FIFO = 0. This interrupt is asserted when the Non-periodic TxFIFO is either half or completely empty, and there is space for at least one entry to be written to the Non-periodic Transmit Request Queue. The half or completely empty status is determined by the Non-periodic TxFIFO Empty Level bit in the Core AHB Configuration register	Host and Device	1'b1	RO

	(GAHBCFG.NPTxFEmpLvl).			
4	RxFIFO Non-Empty (RxFLvl) Indicates that there is at least one packet pending to be read from the RxFIFO.	Host and Device	1'b0	RO
3	Start of (micro)Frame (Sof) In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF (HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt. In Device mode, in the core sets this bit to indicate that an SOF token has been received on the USB. The application can read the Device Status register to get the current (micro)frame number. This interrupt is seen only when the core is operating at either HS or FS. **Note:** This register may return 1'b1 if read immediately after power on reset. If the register bit reads 1'b1 immediately after power on reset it does not indicate that an SOF has been sent (in case of host mode) or SOF has been received (in case of device mode). The read value of this interrupt is valid only after a valid connection between host and device is established. If the bit is set after power on reset the application can clear the bit.	Host and Device	1'b0	R_SS_WC
2	OTG Interrupt (OTGInt) The core sets this bit to indicate an OTG protocol event. The application must read the OTG Interrupt Status (GOTGINT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the GOTGINT register to clear this bit.	Host and Device	1'b0	RO
1	Mode Mismatch Interrupt (ModeMis) The core sets this bit when the application is trying to access: ▪ A Host mode register, when the core is operating in Device mode ▪ A Device mode register, when the core is operating in Host mode The register access is completed on the AHB with an OKAY response, but is ignored by the core internally and does not affect the operation of the core.	Host and Device	1'b0	R_SS_WC
0	Current Mode of Operation (CurMod) Indicates the current mode. ▪ 1'b0: Device mode ▪ 1'b1: Host mode	Host and Device	1'b0	RO

25.5.3.7 Interrupt Mask Register (GINTMSK)

✧ Offset: 018h

This register works with the "Interrupt Register (GINTSTS)" to interrupt the application. When an interrupt bit is masked, the interrupt associated with that bit is not generated. However, the GINTSTS register bit corresponding to that interrupt is still set.

✧ Mask interrupt: 1'b0

✧ Unmask interrupt: 1'b1

Table 25-14 Interrupt Mask Register: GINTMSK

Field	Description	Mode	Reset	Access
31	Resume/Remote Wakeup Detected Interrupt Mask (WkUpIntMsk)	Host and Device	1'b0	R_W
30	Session Request/New Session Detected Interrupt Mask (SessReqIntMsk)	Host and Device	1'b0	R_W
29	Disconnect Detected Interrupt Mask (DisconnIntMsk)	Host Only	1'b0	R_W
28	Connector ID Status Change Mask (ConIDStsChngMsk)	Host and Device	1'b0	R_W
27	LPM Transaction Received Interrupt Mask (LPM_IntMsk)		1'b0	
26	Periodic TxFIFO Empty Mask (PTxFEmpMsk)	Host Only	1'b1	R_W
25	Host Channels Interrupt Mask (HChIntMsk)	Host Only	1'b0	R_W
24	Host Port Interrupt Mask (PrtIntMsk)	Host Only	1'b0	R_W
23	Reset Detected Interrupt Mask (ResetDetMsk)	Device Only	1'b0	R_W
22	Data Fetch Suspended Mask (FetSuspMsk)	Device Only	1'b0	R_W
21	Incomplete Periodic Transfer Mask (incomplPMsk) Incomplete Isochronous OUT Transfer Mask (incompISOOUTMsk)	Host only Device only	1'b0	R_W
20	Incomplete Isochronous IN Transfer Mask (incompISOINMsk) This bit is enabled only when device periodic endpoints are enabled in Dedicated TxFIFO mode.	Device only	1'b0	R_W
19	OUT Endpoints Interrupt Mask (OEPIntMsk)	Device only	1'b0	R_W
18	IN Endpoints Interrupt Mask (IEPIntMsk)	Device	1'b0	R_W

		only		
17	Endpoint Mismatch Interrupt Mask (EPMisMsk)	Device only	1'b0	R_W
16	Restore Done Interrupt Mask (RstrDoneIntMsk) This field is valid only when Hibernation feature is enabled (OTG_EN_PWROPT=2).	Host and Device mode	1'b0	R_W
15	End of Periodic Frame Interrupt Mask (EOPFMsk)	Device only	1'b0	R_W
14	Isochronous OUT Packet Dropped Interrupt Mask (ISOOutDropMsk)	Device only	1'b0	R_W
13	Enumeration Done Mask (EnumDoneMsk)	Device only	1'b0	R_W
12	USB Reset Mask (USBRstMsk)	Device only	1'b0	R_W
11	USB Suspend Mask (USBSuspMsk)	Device only	1'b0	R_W
10	Early Suspend Mask (ErlySuspMsk)	Device only	1'b0	R_W
9	I^2C Interrupt Mask (I2CIntMsk)	Host and Device	1'b0	R_W
8	ULPI Carkit Interrupt Mask (ULPICKINTMsk) I^2C Carkit Interrupt Mask (I2CCKINTMsk)	Host and Device	1'b0	R_W
7	Global OUT NAK Effective Mask (GOUTNakEffMsk)	Device only	1'b0	R_W
6	Global Non-periodic IN NAK Effective Mask (GINNakEffMsk)	Device only	1'b0	R_W
5	Non-periodic TxFIFO Empty Mask (NPTxFEmpMsk)	Host and Device	1'b1	R_W
4	Receive FIFO Non-Empty Mask (RxFLvlMsk)	Host and Device	1'b0	R_W
3	Start of (micro)Frame Mask (SofMsk)	Host and Device	1'b0	R_W
2	OTG Interrupt Mask (OTGIntMsk)	Host and Device	1'b0	R_W
1	Mode Mismatch Interrupt Mask (ModeMisMsk)	Host and Device	1'b0	R_W
0	Reserved	Host and Device		RO

25.5.3.8 Receive Status Debug Read/Status Read and Pop Registers (GRXSTSR/GRXSTSP)

✧ Offset for Read: 01Ch

✧ Offset for Pop: 020h

A read to the Receive Status Debug Read register returns the contents of the top of the Receive FIFO. A read to the Receive Status Read and Pop register additionally pops the top data entry out of the RxFIFO.

The receive status contents must be interpreted differently in Host and Device modes. The core ignores the receive status pop/read when the receive FIFO is empty and returns a value of 32'h0000_0000. The application must only pop the Receive Status FIFO when the Receive FIFO Non-Empty bit of the Core Interrupt register (GINTSTS.RxFLvl) is asserted.

Note:

- Use of these fields vary based on whether the HS OTG core is functioning as a host or a device.
- Do not read this register's reset value before configuring the core because the read value is "X" in the simulation.

Table 25-15 shows Host mode.

Table 25-15 Host Mode Receive Status Debug Read/Status Read and Pop Registers: GRXSTSR/GRXSTSP

Field	Description	Reset	Access
31:21	Reserved		RO
20:17	Packet Status (PktSts) ▪ 4'b0010: IN data packet received ▪ 4'b0011: IN transfer completed (triggers an interrupt) ▪ 4'b0101: Data toggle error (triggers an interrupt) ▪ 4'b0111: Channel halted (triggers an interrupt) ▪ Others: Reserved	4'b0	RO
16:15	Data PID (DPID) Indicates the Data PID of the received packet	2'b0	RO
14:4	Byte Count (BCnt) Indicates the byte count of the received IN data packet.	11'b0	RO
3:0	Channel Number (ChNum) Indicates the channel number to which the current received packet belongs.	4'h0	RO

Table 25-16 shows Device mode

Table 25-16 Device Mode Receive Status Debug Read/Status Read and Pop Registers: GRXSTSR/GRXSTSP

Field	Description	Reset	Acces
31:25	Reserved		RO
24:21	Frame Number (FN) This is the least significant 4 bits of the (micro)frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.	4'b0	RO

20:17	Packet Status (PktSts) Indicates the status of the received packet ▪ 4'b0001: Global OUT NAK (triggers an interrupt) ▪ 4'b0010: OUT data packet received ▪ 4'b0011: OUT transfer completed (triggers an interrupt) ▪ 4'b0100: SETUP transaction completed (triggers an interrupt) ▪ 4'b0110: SETUP data packet received ▪ Others: Reserved	4'b0	RO
16:15	Data PID (DPID) Indicates the Data PID of the received OUT data packet ▪ 2'b00: DATA0 ▪ 2'b10: DATA1 ▪ 2'b01: DATA2 ▪ 2'b11: MDATA	2'b0	RO
14:4	Byte Count (BCnt) Indicates the byte count of the received data packet.	11'h0	RO
3:0	Endpoint Number (EPNum) Indicates the endpoint number to which the current received packet belongs.	4'b0	RO

25.5.3.9 Receive FIFO Size Register (GRXFSIZ)

✧ Offset: 024h

The application can program the RAM size that must be allocated to the RxFIFO.

Table 25-17 Receive FIFO Size Register: GRXFSIZ

Field	Description	Reset	Access
31:16	Reserved		RO
15:0	RxFIFO Depth (RxFDep) This value is in terms of 32-bit words. ▪ Minimum value is 16 ▪ Maximum value is 3648.	16'd3648	R_W

25.5.3.10 Non-Periodic Transmit FIFO Size Register (GNPTXFSIZ)

✧ Offset: 028h

The application can program the RAM size and the memory start address for the Non-periodic TxFIFO.

Note: The fields of this register change, depending on host or device mode.

Table 25-18 Non-Periodic Transmit FIFO Size Register: GNPTXFSIZ (Host Mode and Device Shared FIFO Mode)

Field	Description	Reset	Access
31:16	Non-periodic TxFIFO Depth (NPTxFDep) For host mode, this field is always valid. For Device mode, this field is invalid. This value is in terms of 32-bit words. ▪ Minimum value is 16 ▪ Maximum value is 3648	Configurable	RO/R_W
15:0	Non-periodic Transmit RAM Start Address (NPTxFStAddr) For host mode, this field is always valid. This field contains the memory start address for Non-periodic Transmit FIFO RAM. ▪ The application can write a new value in this field. Programmed values must not exceed the power-on value set	Configurable	R_W

Table 25-19 Non-Periodic Transmit FIFO Size Register: GNPTXFSIZ (Device Dedicated FIFO Mode)

Field	Description	Reset	Acces
31:16	IN Endpoint TxFIFO 0 Depth (INEPTxF0Dep)This field is valid only for Device mode . This value is in terms of 32-bit words. ▪ Minimum value is 16 ▪ Maximum value is 3648	Configurable	RO/R_W
15:0	IN Endpoint FIFO0 Transmit RAM Start Address (INEPTxF0StAddr) For Device mode, this field is valid. This field contains the memory start address for IN Endpoint Transmit FIFO# 0. Programmed values must not exceed the power-on value set.	Configurable	RO/R_W

25.5.3.11 Non-Periodic Transmit FIFO/Queue Status Register (GNPTXSTS)

✧ Offset: 02Ch

In Device mode, this register is valid only in Shared FIFO operation.

This read-only register contains the free space information for the Non-periodic TxFIFO and the Non-periodic Transmit Request Queue.

Table 25-20 Non-Periodic Transmit FIFO/Queue Status Register: GNPTXSTS

Field	Description	Reset	Access
31:24	Top of the Non-periodic Transmit Request Queue (NPTxQTop) Entry in the Non-periodic Tx Request Queue that is currently being	7'b0	RO

	processed by the MAC. ▪ Bits [30:27]: Channel/endpoint number ▪ Bits [26:25]: ▪ 2'b00: IN/OUT token – 2'b01: Zero-length transmit packet (device IN/host OUT) – 2'b10: PING/CSPLIT token – 2'b11: Channel halt command ▪ Bit [24]: Terminate (last entry for selected channel/endpoint)		
23:16	Non-periodic Transmit Request Queue Space Available (NPTxQSpcAvail) Indicates the amount of free space available in the Non-periodic Transmit Request Queue. This queue holds both IN and OUT requests in Host mode. Device mode has only IN requests. ▪ 8'h0: Non-periodic Transmit Request Queue is full ▪ 8'h1: 1 location available ▪ 8'h2: 2 locations available ▪ n : n locations available (0 ≤n ≤8) ▪ Others: Reserved	Configurable	RO
15:0	Non-periodic TxFIFO Space Avail (NPTxFSpcAvail) Indicates the amount of free space available in the Non-periodic TxFIFO. Values are in terms of 32-bit words. ▪ 16'h0: Non-periodic TxFIFO is full ▪ 16'h1: 1 word available ▪ 16'h2: 2 words available ▪ 16'h*n*: *n* words available (where 0 ≤n ≤32,768) ▪ 16'h8000: 32,768 words available ▪ Others: Reserved	Configurable	RO

25.5.3.12 User HW Config1 Register (GHWCFG1)

✧ Offset: 044

This register contains the logical endpoint direction(s) selected.

Table 25-21 User HW Config1 Register: GHWCFG1

Field	Description	Reset	Access
31:0	Endpoint Direction (epdir) This 32-bit field uses two bits per endpoint to determine the endpoint direction. Endpoint ▪ Bits [31:30]: Endpoint 15 direction ▪ Bits [29:28]: Endpoint 14 direction ...	Configurable	RO

	▪ Bits [3:2]: Endpoint 1 direction ▪ Bits[1:0]: Endpoint 0 direction (always BIDIR) Direction ▪ 2'b00: BIDIR (IN and OUT) endpoint ▪ 2'b01: IN endpoint ▪ 2'b10: OUT endpoint ▪ 2'b11: Reserved This field is configured using "Name: OTG_EP_DIR_1(n)".		

25.5.3.13 User HW Config2 Register (GHWCFG2)

✧ Offset: 048h

This register contains configuration options selected.

Table 25-22 User HW Config2 Register: GHWCFG2

Field	Description	Reset	Access
31	OTG_ENABLE_IC_USB IC_USB mode specified for mode of operation (parameter OTG_ENABLE _IC_USB) in coreConsultant. To choose IC_USB_MODE, both OTG_FSPHY_INTERFACE and OTG_ENABLE_IC_USB must be 1.	Configurable	RO
30:26	Device Mode IN Token Sequence Learning Queue Depth (TknQDepth) Range: 0-30 This field is configured using "Name: OTG_TOKEN_QUEUE_DEPTH".	Configurable	RO
25:24	Host Mode Periodic Request Queue Depth (PTxQDepth) ▪ 2'b00: 2 ▪ 2'b01: 4 ▪ 2'b10: 8 ▪ 2'b11: 16 This field is configured using parameter "Name: OTG_PERIO_TX_QUEUE_DEPTH"	Configurable	RO
23:22	Non-periodic Request Queue Depth (NPTxQDepth) ▪ 2'b00: 2 ▪ 2'b01: 4 ▪ 2'b10: 8 ▪ Others: Reserved This field is configured using parameter "Name: OTG_NPERIO_TX_QUEUE_DEPTH".	Configurable	RO
21	Reserved		RO
20	Multi Processor Interrupt Enabled (MultiProcIntrpt) ▪ 1'b0: No	Configurable	RO

	▪ 1'b1: Yes This field is configured using parameter "Name: OTG_MULTI_PROC_INTRPT".		
19	Dynamic FIFO Sizing Enabled (DynFifoSizing) ▪ 1'b0: No ▪ 1'b1: Yes This field is configured using parameter "Name: OTG_DFIFO_DYNAMIC".	Configurable	RO
18	Periodic OUT Channels Supported in Host Mode (PerioSupport) ▪ 1'b0: No ▪ 1'b1: Yes This field is configured using parameter "Name: OTG_EN_PERIO_HOST".	Configurable	RO
17:14	Number of Host Channels (NumHstChnl) Indicates the number of host channels supported by the core in Host mode. The range of this field is 0-15: 0 specifies 1 channel, 15 specifies 16 channels. This field is configured using parameter "Name: OTG_NUM_HOST_CHAN".	Configurable	RO
13:10	Number of Device Endpoints (NumDevEps) Indicates the number of device endpoints supported by the core in Device mode in addition to control endpoint 0. The range of this field is 1-15. This field is configured using parameter "Name: OTG_NUM_EPS".	Configurable	RO
9:8	Full-Speed PHY Interface Type (FSPhyType) ▪ 2'b00: Full-speed interface not supported ▪ 2'b01: Dedicated full-speed interface ▪ 2'b10: FS pins shared with UTMI+ pins ▪ 2'b11: FS pins shared with ULPI pins This field is configured using parameter "Name: OTG_FSPHY_INTERFACE".	Configurable	RO
7:6	High-Speed PHY Interface Type (HSPhyType) ▪ 2'b00: High-Speed interface not supported ▪ 2'b01: UTMI+ ▪ 2'b10: ULPI ▪ 2'b11: UTMI+and ULPI This field is configured using parameter "Name: OTG_HSPHY_INTERFACE".	Configurable	RO
5	Point-to-Point (SingPnt) 1'b0: Multi-point application (hub and split support) 1'b1: Single-point application (no hub and no split support) This field is configured using parameter "Name: OTG_SINGLE_POINT".	Configurable	RO

4:3	Architecture (OtgArch) ▪ 2'b00: Slave-Only ▪ 2'b01: External DMA ▪ 2'b10: Internal DMA ▪ Others: Reserved This field is configured using parameter "Name: OTG_ARCHITECTURE".	Configurable	RO
2:0	Mode of Operation (OtgMode) ▪ 3'b000: HNP- and SRP-Capable OTG (Host and Device) ▪ 3'b001: SRP-Capable OTG (Host and Device) ▪ 3'b010: Non-HNP and Non-SRP Capable OTG (Host and Device) ▪ 3'b011: SRP-Capable Device ▪ 3'b100: Non-OTG Device ▪ 3'b101: SRP-Capable Host ▪ 3'b110: Non-OTG Host ▪ Others: Reserved This field is configured using parameter "Name: OTG_MODE".	Configurable	RO

25.5.3.14 User HW Config3 Register (GHWCFG3)

✧ Offset: 04Ch

This register contains the configuration options selected.

Table 25-23 User HW Config3 Register: GHWCFG3

Field	Description	Reset	Access
31:16	DFIFO Depth (DfifoDepth minus EP_LOC_CNT) This value is in terms of 32-bit words. ▪ Minimum value is 32 ▪ Maximum value is 32,768 This field is configured using parameter "Name: OTG_DFIFO_DEPTH" and EP_LOC_CNT. For more information on EP_LOC_CNT, see "Endpoint Information Controller (EPINFO_CTL)".	Configurable	RO
15	OTG_ENABLE_LPM LPM mode specified for Mode of Operation (parameter OTG_ENABLE_LPM) in coreConsultant configuration.	Configurable	RO
14	OTG_BC_SUPPORT This bit indicates the HS OTG controller support for Battery Charger. ▪ 0 - No Battery Charger Support ▪ 1 - Battery Charger support present.	Configurable	RO
13	OTG_ENABLE_HSIC HSIC mode specified for Mode of Operation in coreConsultant		RO

	configuration (parameter OTG_ENABLE_HSIC). **Value Range:** 0-1 ▪ 1: HSIC-capable with shared UTMI PHY interface ▪ 0: Non-HSIC-capable **Parameter Name:** OTG_ENABLE_HSIC **Dependencies:** OTG_HSPHY_INTERFACE **Default:** Non-HSIC-capable		
12	OTG_ADP_SUPPORT This bit indicates whether ADP logic is present within or external to the HS OTG controller ▪ 0 - No ADP logic present with HS OTG controller ▪ 1- ADP logic is present along with HS OTG controller.		RO
11	Reset Style for Clocked always Blocks in RTL (RstType) ▪ 1'b0: Asynchronous reset is used in the core ▪ 1'b1: Synchronous reset is used in the core This field is configured using parameter "Name: OTG_SYNC_RESET_TYPE".	Configurable	RO
10	Optional Features Removed (OptFeature) Indicates whether the User ID register, GPIO interface ports, and SOF toggle and counter ports were removed for gate count optimization by enabling Remove Optional Features? during coreConsultant configuration. ▪ 1'b0: No ▪ 1'b1: Yes This field is configured using parameter "Name: OTG_RM_OPT_FEATURES".	Configurable	RO
9	Vendor Control Interface Support (VndctlSupt) ▪ 1'b0: Vendor Control Interface is not available on the core. ▪ 1'b1: Vendor Control Interface is available. This field is configured using parameter "Name: OTG_VENDOR_CTL_INTERFACE".	Configurable	RO
8	I^2C Selection (I2CIntSel) ▪ 1'b0: I 2C Interface is not available on the core. ▪ 1'b1: I 2C Interface is available on the core. This field is configured using parameter "Name: OTG_I2C_INTERFACE".	Configurable	RO
7	OTG Function Enabled (OtgEn) The application uses this bit to indicate the DWC_otg core's OTG capabilities. ▪ 1'b0: Not OTG capable ▪ 1'b1: OTG Capable This field is configured using parameter "Name: OTG_MODE".	Configurable	RO

6:4	Width of Packet Size Counters (PktSizeWidth) ▪ 3'b000: 4 bits ▪ 3'b001: 5 bits ▪ 3'b010: 6 bits ▪ 3'b011: 7 bits ▪ 3'b100: 8 bits ▪ 3'b101: 9 bits ▪ 3'b110: 10 bits ▪ Others: Reserved This field is configured using parameter "Name: OTG_PACKET_COUNT_WIDTH".	Configurable	RO
3:0	Width of Transfer Size Counters (XferSizeWidth) ▪ 4'b0000: 11 bits ▪ 4'b0001: 12 bits ... ▪ 4'b1000: 19 bits ▪ Others: Reserved This field is configured using parameter "Name: OTG_TRANS_COUNT_WIDTH".	Configurable	RO

25.5.3.15 User HW Config4 Register (GHWCFG4)

✧ Offset: 050h

This register contains the configuration options selected.

Note: Bit [31] is available only when Scatter/Gather DMA mode is enabled. When Scatter/Gather DMA mode is disabled, this field is reserved.

Table 25-24 User HW Config4 Register: GHWCFG4

Field	Description	Reset	Access
31	Scatter/Gather DMA ▪ 1'b1: Dynamic configuration This field is configured using parameter "Name: OTG_EN_DESC_DMA" .	Configurable	RO
30	Scatter/Gather DMA configuration ▪ 1'b0: Non-Scatter/Gather DMA configuration ▪ 1'b1: Scatter/Gather DMA configuration This field is configured using parameter "Name: OTG_EN_DESC_DMA".	Configurable	RO
29:26	Number of Device Mode IN Endpoints Including Control Endpoints (INEps) Range 0 -15 ▪ 0:1 IN Endpoint ▪ 1:2 IN Endpoints	Configurable	RO

	.... ▪ 15:16 IN Endpoints This field is configured using parameter "Name: OTG_NUM_IN_EPS".		
25	Enable Dedicated Transmit FIFO for device IN Endpoints (DedFifoMode) ▪ 1'b0: Dedicated Transmit FIFO Operation not enabled. ▪ 1'b1: Dedicated Transmit FIFO Operation enabled. This field is configured using parameter "Name: OTG_EN_DED_TX_FIFO".	Configurable	RO
24	session_end Filter Enabled (SessEndFltr) ▪ 1'b0: No filter ▪ 1'b1: Filter This field is configured using parameter "Name: OTG_EN_SESSIONEND_FILTER"	Configurable	RO
23	"b_valid" Filter Enabled (BValidFltr) ▪ 1'b0: No filter ▪ 1'b1: Filter This field is configured using parameter "Name: OTG_EN_B_VALID_FILTER".	Configurable	RO
22	"a_valid" Filter Enabled (AValidFltr) ▪ 1'b0: No filter ▪ 1'b1: Filter This field is configured using parameter "Name: OTG_EN_A_VALID_FILTER".	Configurable	RO
21	"vbus_valid" Filter Enabled (VBusValidFltr) ▪ 1'b0: No filter ▪ 1'b1: Filter This field is configured using parameter "Name: OTG_EN_VBUSVALID_FILTER".	Configurable	RO
20	"iddig" Filter Enable (IddgFltr) ▪ 1'b0: No filter ▪ 1'b1: Filter This field is configured using parameter "Name: OTG_EN_IDDIG_FILTER".	Configurable	RO
19:16	Number of Device Mode Control Endpoints in Addition to Endpoint 0 (NumCtlEps) Range: 0-15 This field is configured using parameter "Name: OTG_NUM_CRL_EPS".	Configurable	RO
15:14	UTMI+ PHY/ULPI-to-Internal UTMI+ Wrapper Data Width (PhyDataWidth) When a ULPI PHY is used, an internal wrapper converts ULPI to UTMI+. ▪ 2'b00: 8 bits	Configurable	RO

	▪ 2'b01: 16 bits ▪ 2'b10: 8/16 bits, software selectable ▪ Others: Reserved This field is configured using parameter "Name: OTG_HSPHY_DWIDTH".		
13:6	Reserved		RO
6	Enable Hibernation ▪ 1'b0: Hibernation feature not enabled ▪ 1'b1: Hibernation feature enabled This field is configured using parameter "Name: OTG_EN_PWROPT".	Configurable	RO
5	Minimum AHB Frequency Less Than 60 MHz (AhbFreq) ▪ 1'b0: No ▪ 1'b1: Yes This field is configured using parameter "Name: OTG_MIN_AHB_FREQ_LESSTHAN_60".	Configurable	RO
4	Enable Partial Power Down ▪ 1'b0: Partial Power Down Not Enabled ▪ 1'b1: Partial Power Down Enabled This field is configured using parameter "Name: OTG_EN_PWROPT".	Configurable	RO
3:0	Number of Device Mode Periodic IN Endpoints (NumDevPerioEps) Range: 0-15 This field is configured using parameter "Name: OTG_NUM_PERIO_EPS".	Configurable	RO

25.5.3.16 DFIFO Software Config Register (GDFIFOCFG)

✧ Offset: 05Ch

Table 25-25 Global DFIFO Software Config Register: GDFIFOCFG

Field	Description	Mode	Reset	Access
31:16	EPInfoBaseAddr This field provides the start address of the EP info controller.	Host and Device	EPINFO_BASEADDR	R_W
15:0	GDFIFOCfg This field is for dynamic programming of the DFIFO Size. This value takes effect only when the application programs a non zero value to this register. The value programmed must conform to the guidelines described in "FIFO RAM Allocation" in the Programming Guide. The DWC_otg core does not have any corrective logic if the FIFO sizes are programmed incorrectly.	Host and Device	OTG_DFIFO_DEPTH	R_W

25.5.3.17 Host Periodic Transmit FIFO Size Register (HPTXFSIZ)

✧ Offset: 100h

This register holds the size and the memory start address of the Periodic TxFIFO.

Table 25-26 Host Periodic Transmit FIFO Size Register: HPTXFSIZ

Field	Description	Reset	Access
31:16	Host Periodic TxFIFO Depth (PTxFSize) This value is in terms of 32-bit words. ▪ Minimum value is 16 ▪ Maximum value is 3648	Configurable	R_W
15:0	Host Periodic TxFIFO Start Address (PTxFStAddr) The power-on reset value of this register is the sum of the Largest Rx Data FIFO Depth and Largest Non-periodic Tx Data FIFO Depth specified .These parameters are: In shared FIFO operation:- ▪ OTG_RX_DFIFO_DEPTH(3648) + OTG_TX_NPERIO_DFIFO_DEPTH(1024) In dedicated FIFO mode:- ▪ OTG_RX_DFIFO_DEPTH(3648) + OTG_TX_HNPERIO_DFIFO_DEPTH(3648)	Configurable	RO/R_W

25.5.3.18 Device Periodic Transmit FIFO-*n* Size Register (DPTXFSIZn)

✧ FIFO_number: 1 ≤ n ≤ 15

✧ Offset: 104h + (FIFO_number - 1) * 04h

This register is valid only in shared FIFO operation (OTG_EN_DED_TX_FIFO = 0).

This register holds the memory start address of each periodic TxFIFO to be implemented in Device mode. Each periodic FIFO holds the data for one periodic IN endpoint. This register is repeated for each periodic FIFO instantiated.

Table 25-27 Device Periodic Transmit FIFO-n Register: DPTXFSIZn

Field	Description	Reset	Access
31:16	Device Periodic TxFIFO Size (DPTxFSize) This value is in terms of 32-bit words. ▪ Minimum value is 4 ▪ Maximum value is 768	Configurable	RO
15:0	Device Periodic TxFIFO RAM Start Address (DPTxFStAddr) This field specifies the start address in the RAM for this periodic FIFO. The power- on reset value of this register is the sum of the Largest Rx Data FIFO Depth, Largest Non-periodic Tx Data FIFO Depth, and all lower numbered Largest Device Mode Periodic Tx Data FIFOn Depth specified .	Configurable	RO/R_W

	The formula used is: OTG_RX_DFIFO_DEPTH + OTG_TX_NPERIO_DFIFO_DEPTH + SUM of OTG_TX_DPERIO_DFIFO_DEPTH_'x' (where x=1 to n-1). When n = 1, the above expression becomes OTG_RX_DFIFO_DEPTH + OTG_TX_NPERIO_DFIFO_DEPTH. If at POR, the calculated value (C) exceeds 65535, then the Reset value becomes Reset Value(A) = (C - 65536).		

25.5.3.19 Device IN Endpoint Transmit FIFO Size Register: (DIEPTXFn)

✧ FIFO_number: 1 ≤ n ≤ 15

✧ Offset: 104h + (FIFO_number - 1) * 04h

This register is valid only in dedicated FIFO mode.

This register holds the size and memory start address of IN endpoint TxFIFOs implemented in Device mode. Each FIFO holds the data for one IN endpoint. This register is repeated for instantiated IN endpoint FIFOs 1 to 15. For IN endpoint FIFO 0 use GNPTXFSIZ register for programming the size and memory start address.

Table 25-28 Device In Endpoint Transmit FIFO Size Register: (DIEPTXFn)

Field	Description	Reset	Access
31:16	IN Endpoint TxFIFO Depth (INEPnTxFDep) This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 3648	Configurable	R_W
15:0	IN Endpoint FIFO*n* Transmit RAM Start Address (INEPnTxFStAddr) This field contains the memory start address for IN endpoint Transmit FIFOn (0 < n <= 15). The power-on reset value of this register is calculated according to the following formula: OTG_RX_DFIFO_DEPTH + SUM of OTG_TX_DINEP_DFIFO_DEPTH_'x' (where x = 0 to n - 1) If at POR the calculated value (C) exceeds 65535, then the Reset value becomes Reset Value(A) = (C - 65536). Example: If start address of IN endpoint FIFO 1 is OTG_RX_DFIFO_DEPTH + OTG_TX_DINEP_DFIFO_DEPTH_ 0 and start address of IN endpoint FIFO 2 is OTG_RX_DFIFO_DEPTH + OTG_TX_DINEP_DFIFO_DEPTH_ 0 + OTG_TX_DINEP_DFIFO_DEPTH_ 1.	Configurable	RO/R_W

25.5.4 Host Mode Registers

These registers affect the operation of the core in the Host mode. Host mode registers must not be

accessed in Device mode, as the results are undefined. Host Mode registers can be categorized as follows:

25.5.4.1 Host Configuration Register (HCFG)

✧ Offset: 400h

This register configures the core after power-on. Do not make changes to this register after initializing the host.

Table 25-29 Host Configuration Register: HCFG

Field	Description	Reset	Access
31	Mode Change Ready Timer Enable (ModeChTimEn) This bit is used to enable or disable the host core to wait for 200 PHY clock cycles at the end of Resume to change the opmode signal to the PHY to 00 after Suspend or LPM. ▪ 1'b0: The Host core waits for either 200 PHY clock cycles or a linestate of SE0 at the end of resume to change the opmode from 2'b10 to 2'b00. ▪ 1'b1: The Host core waits only for a linestate of SE0 at the end of resume to change the opmode from 2'b10 to 2'b00.	1'b0	R_W
30:27	Reserved		RO
26	Enable Periodic Scheduling (PerSchedEna) Applicable in Scatter/Gather DMA mode only. Enables periodic scheduling within the core. Initially, the bit is reset. The core will not process any periodic channels. As soon as this bit is set, the core will get ready to start scheduling periodic channels. In non Scatter/Gather DMA mode, this bit is reserved.	1'b0	R_W
25:24	Frame List Entries (FrListEn). The value in the register specifies the number of entries in the Frame list. This field is valid only in Scatter/Gather DMA mode. ▪ 2'b00: 8 Entries ▪ 2'b01:16 Entries ▪ 2'b10: 32 Entries ▪ 2'b11: 64 Entries In modes other than Scatter/Gather DMA mode, these bits are reserved.	2'b00	R/W
23	Enable Scatter/gather DMA in Host mode (DescDMA) When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation. NOTE: This bit must be modified only once after a reset. The following combinations are available for programming: ▪ GAHBCFG.DMAEn=0, HCFG.DescDMA=0 => Slave mode	1'b0	R/W

	▪ GAHBCFG.DMAEn=0, HCFG.DescDMA=1 => Invalid ▪ GAHBCFG.DMAEn=1, HCFG.DescDMA=0 => Buffered DMA mode ▪ GAHBCFG.DMAEn=1, HCFG.DescDMA=1 => Scatter/Gather DMA mode In non Scatter/Gather DMA mode, this bit is reserved.		
22:16	Reserved	8'd2	R_W
15:8	Resume Validation Period (ResValid) This field is effective only when HCFG.Ena32KHzS is set. It controls the resume period when the core resumes from suspend. The core counts the ResValid number of clock cycles to detect a valid resume when this is set.	8'd2	R_W
7	Enable 32-KHz Suspend Mode (Ena32KHzS) This bit can only be set if the USB 1.1 Full-Speed Serial Transceiver Interface has been selected. If USB 1.1 Full-Speed Serial Transceiver Interface has not been selected, this bit must be zero. When the USB 1.1 Full-Speed Serial Transceiver Interface is chosen and this bit is set, the core expects the 48-MHz PHY clock to be switched to 32 KHz during a suspend.	1'd0	R_W
6:3	Reserved		RO
2	FS- and LS-Only Support (FSLSSupp) The application uses this bit to control the core's enumeration speed. Using this bit, the application can make the core enumerate as a FS host, even if the connected device supports HS traffic. Do not make changes to this field after initial programming. ▪ 1'b0: HS/FS/LS, based on the maximum speed supported by the connected device ▪ 1'b1: FS/LS-only, even if the connected device can support HS	1'b0	R_W
1:0	FS/LS PHY Clock Select (FSLSPclkSel) When the core is in FS Host mode ▪ 2'b00: Internal PHY clock is running at 30/60 MHZ (For UTMI+/ULPI PHY Interfaces) ▪ 2'b01: Internal PHY clock is running at 48MHZ (For 1.1 FS transceiver Interface) ▪ Others: Reserved When the core is in LS Host mode ▪ 2'b00: Internal PHY clock is running at 30/60 MHZ (For UTMI+/ULPI PHY Interfaces) ▪ 2'b10: Internal PHY clock is running at 6 MHZ and the external clock is running at 48MHZ. When you select a 6 MHz clock during LS Mode, you must do a soft reset (for 1.1 FS transceiver Interface) ▪ Others: Reserved	2'b0	R_W

	Notes: ▪ When Core in FS mode, the internal and external clocks have the same frequency. ▪ When Core in LS mode, – If FSLSPclkSel = 2'b00: Internal and external clocks have the same frequency – If FSLSPclkSel = 2'b10: Internal clock is divided by eight version of external 48 MHz clock (utmifs_clk).		

25.5.4.2 Host Frame Interval Register (HFIR)

✧ Offset: 404h

This register stores the frame interval information for the current speed to which the DWC_otg core has enumerated.

Table 25-30 Host Frame Interval Register: HFIR

Field	Description	Reset	Access
31:17	Reserved		RO
16	Reload Control (HFIRRldCtrl) This bit allows dynamic reloading of the HFIR register during runtime. ▪ 1'b0: The HFIR cannot be reloaded dynamically ▪ 1'b1: the HFIR can be dynamically reloaded during runtime. This bit needs to be programmed during initial configuration and its value must not be changed during runtime.	1'b0	R_W
15:0	Frame Interval (FrInt) The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro-SOFs (HS) or Keep-Alive tokens (LS). This field contains the number of PHY clocks that constitute the required frame interval. The default value set in this field for a FS operation when the PHY clock frequency is 60 MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been set. If no value is programmed, the core calculates the value based on the PHY clock specified in the FS/LS PHY Clock Select field of the Host Configuration register (HCFG.FSLSPclkSel). ▪ 125 μs * (PHY clock frequency for HS) ▪ 1 ms * (PHY clock frequency for FS/LS)	16'd60000	R_W

25.5.4.3 Host Frame Number/Frame Time Remaining Register (HFNUM)

✧ Offset: 408h

This register indicates the current frame number. It also indicates the time remaining (in terms of the number of

PHY clocks) in the current (micro)frame.

Table 25-31 Host Frame Number/Frame Time Remaining Register: HFNUM

Field	Description	Reset	Access
31:16	Frame Time Remaining (FrRem) Indicates the amount of time remaining in the current microframe (HS) or frame (FS/LS), in terms of PHY clocks. This field decrements on each PHY clock. When it reaches zero, this field is reloaded with the value in the Frame Interval register and a new SOF is transmitted on the USB.	16'h0	RO
15:0	Frame Number (FrNum) This field increments when a new SOF is transmitted on the USB, and is reset to 0 when it reaches 16'h3FFF.	16'h3FFF	RO

25.5.4.4 Host Periodic Transmit FIFO/Queue Status Register (HPTXSTS)

✧ Offset: 410h

This read-only register contains the free space information for the Periodic TxFIFO and the Periodic Transmit Request Queue.

Table 25-32 Host Periodic Transmit FIFO/Queue Status Register: HPTXSTS

Field	Description	Reset	Access
31:24	Top of the Periodic Transmit Request Queue (PTxQTop) This indicates the entry in the Periodic Tx Request Queue that is currently being processes by the MAC. This register is used for debugging. ▪ Bit [31]: Odd/Even (micro)frame – 1'b0: send in even (micro)frame – 1'b1: send in odd (micro)frame ▪ Bits [30:27]: Channel/endpoint number ▪ Bits [26:25]: Type – 2'b00: IN/OUT – 2'b01: Zero-length packet – 2'b10: CSPLIT – 2'b11: Disable channel command ▪ Bit [24]: Terminate (last entry for the selected channel/endpoint)	8'h0	RO
23:16	Periodic Transmit Request Queue Space Available (PTxQSpcAvail) Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests. ▪ 8'h0: Periodic Transmit Request Queue is full ▪ 8'h1: 1 location available ▪ 8'h2: 2 locations available	Configurable	RO

	▪ n : n locations available (0 ≤n ≤16) ▪ Others: Reserved		
15:0	Periodic Transmit Data FIFO Space Available (PTxFSpcAvail) Indicates the number of free locations available to be written to in the Periodic TxFIFO. Values are in terms of 32-bit words ▪ 16'h0: Periodic TxFIFO is full ▪ 16'h1: 1 word available ▪ 16'h2: 2 words available ▪ 16'hn: n words available (where 0 ≤n ≤32,768) ▪ 16'h8000: 32,768 words available ▪ Others: Reserved	Configurable	RO

25.5.4.5 Host All Channels Interrupt Register (HAINT)

✧ Offset: 414h

When a significant event occurs on a channel, the Host All Channels Interrupt register interrupts the application using the Host Channels Interrupt bit of the Core Interrupt register (GINTSTS.HChInt). There is one interrupt bit per channel, up to a maximum of 16 bits. Bits in this register are set and cleared when the application sets and clears bits in the corresponding Host Channel-n Interrupt register.

Table 25-33 Host All Channels Interrupt Register: HAINT

Field	Description	Reset	Access
31:16	Reserved		RO
15:0	Channel Interrupts (HAINT) One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15	16'h0	RO

25.5.4.6 Host All Channels Interrupt Mask Register (HAINTMSK)

✧ Offset: 418h

The Host All Channel Interrupt Mask register works with the Host All Channel Interrupt register to interrupt the application when an event occurs on a channel. There is one interrupt mask bit per channel, up to a maximum of 16 bits.

✧ Mask interrupt: 1'b0

✧ Unmask interrupt: 1'b1

Table 25-34 Host All Channels Interrupt Mask Register: HAINTMSK

Field	Description	Reset	Access
31:16	Reserved		RO
15:0	Channel Interrupt Mask (HAINTMsk) One bit per channel: Bit 0 for channel 0, bit 15 for channel 15	16'h0	R_W

25.5.4.7 Host Port Control and Status Register (HPRT)

✧ Offset: 440h

This register is available only in Host mode. Currently, the OTG Host supports only one port.

A single register holds USB port-related information such as USB reset, enable, suspend, resume, connect status, and test mode for each port. The R_SS_WC bits in this register can trigger an interrupt to the application through the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt). On a Port Interrupt, the application must read this register and clear the bit that caused the interrupt. For the R_SS_WC bits, the application must write a 1 to the bit to clear the interrupt.

Table 25-35 Host Port Control and Status Register: HPRT

Field	Description	Reset	Access
31:19	Reserved		RO
18:17	Port Speed (PrtSpd) Indicates the speed of the device attached to this port. ▪ 2'b00: High speed ▪ 2'b01: Full speed ▪ 2'b10: Low speed ▪ 2'b11: Reserved	2'h0	RO
16:13	Port Test Control (PrtTstCtl) The application writes a nonzero value to this field to put the port into a Test mode, and the corresponding pattern is signaled on the port. ▪ 4'b0000: Test mode disabled ▪ 4'b0001: Test_J mode ▪ 4'b0010: Test_K mode ▪ 4'b0011: Test_SE0_NAK mode ▪ 4'b0100: Test_Packet mode ▪ 4'b0101: Test_Force_Enable ▪ Others: Reserved For more information on moving the DWC_otg host core into test mode, see "Moving the Host Core to Test Mode" on page 89.	4'h0	R_W
12	Port Power (PrtPwr) The application uses this field to control power to this port (write 1'b1 to set to 1'b1 and write 1'b0 to set to 1'b0), and the core can clear this bit on an over current condition. ▪ 1'b0: Power off ▪ 1'b1: Power on	1'b0	RO
11:10	Port Line Status (PrtLnSts) Indicates the current logic level USB data lines ▪ Bit [10]: Logic level of D+ ▪ Bit [11]: Logic level of D-	2'b0	RO
9	Reserved		RO

8	Port Reset (PrtRst) When the application sets this bit, a reset sequence is started on this port. The application must time the reset period and clear this bit after the reset sequence is complete. ▪ 1'b0: Port not in reset ▪ 1'b1: Port in reset To start a reset on the port, the application must leave this bit set for at least the minimum duration mentioned below, as specified in the USB 2.0 specification, Section 7.1.7.5. The application can leave it set for another 10 ms in addition to the required minimum duration, before clearing the bit, even though there is no maximum limit set by the USB standard. ▪ High speed: 50 ms ▪ Full speed/Low speed: 10 ms	1'b0	R_W
7	Port Suspend (PrtSusp) The application sets this bit to put this port in Suspend mode. The core only stops sending SOFs when this is set. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the suspend input pin of the PHY. The read value of this bit reflects the current suspend status of the port. This bit is cleared by the core after a remote wakeup signal is detected or the application sets the Port Reset bit or Port Resume bit in this register or the Resume/Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.WkUpInt or GINTSTS.DisconnInt, respectively). ▪ 1'b0: Port not in Suspend mode ▪ 1'b1: Port in Suspend mode	1'b0	R_WS_SC
6	Port Resume (PrtRes) The application sets this bit to drive resume signaling on the port. The core continues to drive the resume signal until the application clears this bit. If the core detects a USB remote wakeup sequence, as indicated by the Port Resume/Remote Wakeup Detected Interrupt bit of the Core Interrupt register (GINTSTS.WkUpInt), the core starts driving resume signaling without application intervention and clears this bit when it detects a disconnect condition. The read value of this bit indicates whether the core is currently driving resume signaling. ▪ 1'b0: No resume driven ▪ 1'b1: Resume driven When LPM is enabled and the core is in the L1 (Sleep) state, setting this bit results in the following behavior:	1'b0	R_W_SS_SC

	The core continues to drive the resume signal until a pre-determined time specified in the GLPMCFG.HIRD_Thres[3:0] field. If the core detects a USB remote wakeup sequence, as indicated by the Port L1 Resume/Remote L1 Wakeup Detected Interrupt bit of the Core Interrupt register (GINTSTS.L1WkUpInt), the core starts driving resume signaling without application intervention and clears this bit at the end of the resume. The read value of this bit indicates whether the core is currently driving resume signaling. ▪ 1'b0: No resume driven ▪ 1'b1: Resume driven		
5	Port Overcurrent Change (PrtOvrCurrChng) The core sets this bit when the status of the Port Overcurrent Active bit (bit 4) in this register changes.	1'b0	R_SS_WC
4	Port Overcurrent Active (PrtOvrCurrAct) Indicates the overcurrent condition of the port. ▪ 1'b0: No overcurrent condition ▪ 1'b1: Overcurrent condition	1'b0	RO
3	Port Enable/Disable Change (PrtEnChng) The core sets this bit when the status of the Port Enable bit [2] of this register changes.	1'b0	R_SS_WC
2	Port Enable (PrtEna) A port is enabled only by the core after a reset sequence, and is disabled by an overcurrent condition, a disconnect condition, or by the application clearing this bit. The application cannot set this bit by a register write. It can only clear it to disable the port. This bit does not trigger any interrupt to the application. ▪ 1'b0: Port disabled ▪ 1'b1: Port enabled	1'b0	R_SS_SC_WC
1	Port Connect Detected (PrtConnDet) The core sets this bit when a device connection is detected to trigger an interrupt to the application using the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt). The application must write a 1 to this bit to clear the interrupt.	1'b0	R_SS_WC
0	Port Connect Status (PrtConnSts) ▪ 0: No device is attached to the port. ▪ 1: A device is attached to the port.	1'b0	RO

25.5.4.7.1 Moving the Host Core to Test Mode

To move the DWC_otg core to test mode, you must set HPRT.Port Test Control. Complete the

following steps to move the DWC_otg core to test mode:

1. Power on the core.
2. Load the DWC_otg driver.
3. Connect an HS device and enumerate to HS mode.
4. Access the HPRT register to send test packets.
5. Remove the device and connect to fixture (OPT) port.

The DWC_otg host core continues sending out test packets.

6. Test the eye diagram.

25.5.4.8 Host Channel-*n* Characteristics Register (HCCHARn)

✧ Channel_number: 0 ≤n ≤15

✧ Offset: 500h + (Channel_number * 20h)

Table 25-36 Host Channel-n Characteristics Register: HCCHARn

Field	Description	Reset	Access
31	Channel Enable (ChEna) When Scatter/Gather mode is enabled ▪ 1'b0: Indicates that the descriptor structure is not yet ready. ▪ 1'b1: Indicates that the descriptor structure and data buffer with data is setup and this channel can access the descriptor. When Scatter/Gather mode is disabled This field is set by the application and cleared by the OTG host. ▪ 1'b0: Channel disabled ▪ 1'b1: Channel enabled	1'b0	R_WS_SC
30	Channel Disable (ChDis) The application sets this bit to stop transmitting/receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled.	1'b0	R_WS_SC_SS
29	Odd Frame (OddFrm) This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro)frame. This field is applicable for only periodic (isochronous and interrupt) transactions. ▪ 1'b0: Even (micro)frame ▪ 1'b1: Odd (micro)frame This field is not applicable for Scatter/Gather DMA mode and need not be programmed by the application and is ignored by the core.	1'b0	R_W
28:22	Device Address (DevAddr)	7'h0	R_W

	This field selects the specific device serving as the data source or sink.		
21:20	Multi Count (MC) / Error Count (EC) When the Split Enable bit of the Host Channel-n Split Control register (HCSPLTn.SpltEna) is reset (1'b0), this field indicates to the host the number of transactions that must be executed per microframe for this periodic endpoint. For non periodic transfers, this field is used only in DMA mode, and specifies the number packets to be fetched for this channel before the internal DMA engine changes aubitration. ▪ 2'b00: Reserved This field yields undefined results. ▪ 2'b01: 1 transaction ▪ 2'b10: 2 transactions to be issued for this endpoint per microframe ▪ 2'b11: 3 transactions to be issued for this endpoint per microframe When HCSPLTn.SpltEna is set (1'b1), this field indicates the number of immediate retries to be performed for a periodic split transactions on transaction errors. This field must be set to at least 2'b01.	2'b0	R_W
19:18	Endpoint Type (EPType) Indicates the transfer type selected. ▪ 2'b00: Control ▪ 2'b01: Isochronous ▪ 2'b10: Bulk ▪ 2'b11: Interrupt	2'b0	R_W
17	Low-Speed Device (LSpdDev) This field is set by the application to indicate that this channel is communicating to a low-speed device. The application must program this bit when a low speed device is connected to the host through an FS HUB. The HS OTG Host core uses this field to drive the XCVR_SELECT signal to 2'b11 while communicating to the LS Device through the FS hub. **Note:** In a peer to peer setup, the HS OTG Host core ignores this bit even if it is set by the application software.	1'b0	R_W
16	Reserved		RO
15	Endpoint Direction (EPDir) Indicates whether the transaction is IN or OUT. ▪ 1'b0: OUT ▪ 1'b1: IN	1'b0	R_W
14:11	Endpoint Number (EPNum) Indicates the endpoint number on the device serving as the data source or sink.	4'h0	R_W

10:0	Maximum Packet Size (MPS) Indicates the maximum packet size of the associated endpoint.	11'h0	R_W

25.5.4.9 Host Channel-n Split Control Register (HCSPLTn)

✧ Channel_number: 0 ≤n ≤15

✧ Offset: 504h + (Channel_number * 20h)

Table 25-37 Host Channel-n Split Control Register: HCSPLTn

Field	Description	Reset	Access
31	Split Enable (SpltEna) The application sets this field to indicate that this channel is enabled to perform split transactions.	1'b0	R_W
30:17	Reserved		RO
16	Do Complete Split (CompSplt) The application sets this field to request the OTG host to perform a complete split trsnsaction.	1'b0	R_W
15:14	Transaction Position (XactPos) This field is used to determine whether to send all, first, middle, or last payloads with each OUT transaction. ▪ 2'b11: All. This is the entire data payload is of this transaction (which is less than or equal to 188 bytes). ▪ 2'b10: Begin. This is the first data payload of this transaction (which is larger than 188 bytes). ▪ 2'b00: Mid. This is the middle payload of this transaction (which is larger than 188 bytes). ▪ 2'b01: End. This is the last payload of this transaction (which is larger than 188 bytes).	2'h0	R_W
13:7	Hub Address (HubAddr) This field holds the device address of the transaction translator's hub.	7'h0	R_W
6:0	Port Address (PrtAddr) This field is the port number of the recipient transaction translator.	7'h0	R_W

25.5.4.10 Host Channel-*n* Interrupt Register (HCINTn)

✧ Channel_number: 0 ≤n ≤15

✧ Offset: 508h + (Channel_number * 20h)

This register indicates the status of a channel with respect to USB- and AHB-related events. The application must read this register when the Host Channels Interrupt bit of the Core Interrupt register (GINTSTS.HChInt) is set. Before the application can read this register, it must first read the Host All

Channels Interrupt (HAINT) register to get the exact channel number for the Host Channel-n Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the HAINT and GINTSTS registers.

Table 25-38 Host Channel-n Interrupt Register: HCINTn

Field	Description	Reset	Access
31:11	Reserved		RO
13	Descriptor rollover interrupt (DESC_LST_ROLLIntr) This bit is valid only when Scatter/Gather DMA mode is enabled. The core sets this bit when the corresponding channel's descriptor list rolls over. For non Scatter/Gather DMA mode, this bit is reserved.	1'b0	R_SS_WC
12	Excessive Transaction Error (XCS_XACT_ERR) This bit is valid only when Scatter/Gather DMA mode is enabled. The core sets this bit when 3 consecutive transaction errors occurred on the USB bus. XCS_XACT_ERR will not be generated for Isochronous channels. For non Scatter/Gather DMA mode, this bit is reserved.	1'b0	R_SS_WC
11	BNA (Buffer Not Available) Interrupt (BNAIntr) This bit is valid only when Scatter/Gather DMA mode is enabled. The core generates this interrupt when the descriptor accessed is not ready for the Core to process. BNA will not be generated for Isochronous channels. For non Scatter/Gather DMA mode, this bit is reserved.	1'b0	R_SS_WC
10	Data Toggle Error (DataTglErr) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.	1'b0	R_SS_WC
9	Frame Overrun (FrmOvrun) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core	1'b0	R_SS_WC
8	Babble Error (BblErr) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.	1'b0	R_SS_WC
7	Transaction Error (XactErr) Indicates one of the following errors occurred on the USB. ▪ CRC check failure ▪ Timeout ▪ Bit stuff error ▪ False EOP In Scatter/Gather DMA mode, the interrupt due to this bit is not set.	1'b0	R_SS_WC
6	NYET Response Received Interrupt (NYET)	1'b0	R_SS_WC

	In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.		
5	ACK Response Received/Transmitted Interrupt (ACK) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.	1'b0	R_SS_WC
4	NAK Response Received Interrupt (NAK) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.	1'b0	R_SS_WC
3	STALL Response Received Interrupt (STALL) In Scatter/Gather DMA mode, the interrupt due to this bit is masked in the core.	1'b0	R_SS_WC
2	AHB Error (AHBErr) This is generated only in DMA mode when there is an AHB error during AHB read/write. The application can read the corresponding channel's DMA address register to get the error address.	1'b0	R_SS_WC
1	Channel Halted (ChHltd) In non Scatter/Gather DMA mode, it indicates the transfer completed abnormally either because of any USB transaction error or in response to disable request by the application or because of a completed transfer. In Scatter/Gather DMA mode, this indicates that transfer completed due to any of the following ▪ EOL being set in descriptor ▪ AHB error ▪ Excessive transaction errors ▪ In response to disable request by the application ▪ Babble ▪ Stall	1'b0	R_SS_WC
0	Transfer Completed (XferCompl) For Scatter/Gather DMA mode, it indicates that current descriptor processing got completed with IOC bit set in its descriptor. In non Scatter/Gather DMA mode, it indicates that Transfer completed normally without any errors.	1'b0	R_SS_WC

25.5.4.11 Host Channel-*n* Interrupt Mask Register (HCINTMSKn)

✧ Channel_number: 0 ≤n ≤15

✧ Offset: 50Ch + (Channel_number * 20h)

This register reflects the mask for each channel status described in the previous section.

✧ Mask interrupt: 1'b0

✧ Unmask interrupt: 1'b1

Table 25-39 Host Channel-n Interrupt Mask Register: HCINTMSKn

Field	Description	Reset	Access
31:11	Reserved	1'b0	RO
13	Descriptor rollover interrupt Mask register (DESC_LST_ROLLIntrMsk) This bit is valid only when Scatter/Gather DMA mode is enabled. In non Scatter/Gather DMA mode, this bit is reserved.	1'b0	R_W
12	Reserved	1'b0	RO
11	Reserved	1'b0	RO
11	BNA (Buffer Not Available) Interrupt mask register (BNAIntrMsk) This bit is valid only when Scatter/Gather DMA mode is enabled. In non Scatter/Gather DMA mode, this bit is reserved	1'b0	R_W
10	Data Toggle Error Mask (DataTglErrMsk) This bit is not applicable in Scatter/Gather DMA mode.	1'b0	RO
9	Frame Overrun Mask (FrmOvrunMsk) This bit is not applicable in Scatter/Gather DMA mode.	1'b0	R_W
8	Babble Error Mask (BblErrMsk) This bit is not applicable in Scatter/Gather DMA mode.	1'b0	R_W
7	Transaction Error Mask (XactErrMsk) This bit is not applicable in Scatter/Gather DMA mode	1'b0	R_W
6	NYET Response Received Interrupt Mask (NyetMsk) This bit is not applicable in Scatter/Gather DMA mode.	1'b0	R_W
5	ACK Response Received/Transmitted Interrupt Mask (AckMsk) This bit is not applicable in Scatter/Gather DMA mode.	1'b0	R_W
4	NAK Response Received Interrupt Mask (NakMsk) This bit is not applicable in Scatter/Gather DMA mode.	1'b0	R_W
3	STALL Response Received Interrupt Mask (StallMsk) This bit is not applicable in Scatter/Gather DMA mode.		
2	AHB Error Mask (AHBErrMsk) Note: This bit is only accessible when OTG_ARCHITECTURE = 2		
1	Channel Halted Mask (ChHltdMsk)	1'b0	R_W
0	Transfer Completed Mask (XferComplMsk)	1'b0	R_W

25.5.4.12 Host Channel-*n* Transfer Size Register (HCTSIZn)

✧ Channel_number: 0≤n≤15

✧ Offset: 510h + (Channel_number * 20h)

In Scatter/Gather DMA mode, the HCTSIZn register is defined as described inTable 25-40.

Table 25-40 Host Channel-n Transfer Size Register: HCTSIZn

Field	Description	Reset	Access
30:29	PID (Pid) The application programs this field with the type of PID to use for the initial transaction. The host maintains this field for the rest of the transfer. ▪ 2'b00: DATA0 ▪ 2'b01: DATA2 ▪ 2'b10: DATA1 ▪ 2'b11: MDATA (non-control)	2'b00	R_W
28:19	Reserved	1'b0	RO
18:16	Reserved	1'b0	RO
15:8	NTD (Number of Transfer Descriptors) **(Non Isochronous)** This value is in terms of number of descriptors. Maximum number of descriptor that can be present in the list is 64. The values can be from 0 to 63. ▪ 0 - 1 descriptor. ▪ 63 - 64 descriptors This field indicates the total number of descriptors present in that list. The core will wrap around after servicing NTD number of descriptors for that list. **(Isochronous)** This field indicates the number of descriptors present in that list.µframe The possible values for FS are ▪ 1 - 2 descriptors ▪ 3 - 4 descriptors ▪ 7 - 8 descriptors ▪ 15 - 16 descriptors ▪ 31 - 32 descriptors ▪ 63 - 64 descriptors The possible values for HS are ▪ 7 - 8 descriptors ▪ 15 - 16 descriptors ▪ 31 - 32 descriptors ▪ 63 - 64 descriptors ▪ 127 - 128 descriptors ▪ 255 - 256 descriptors	8'h0	R_W
7:0	SCHED_INFO (Schedule information) Every bit in this 8 bit register indicates scheduling for that microframe. Bit 0 indicates scheduling for 1st microframe and bit	8'h0	R_W

	7 indicates scheduling for 8th microframe in that frame. A value of 8'b11111111 indicates that the corresponding interrupt channel is scheduled to issue a token every microframe in that frame. A value of 8'b10101010 indicates that the corresponding interrupt channel is scheduled to issue a token every alternate microframe starting with second microframe. Note that this field is applicable only for periodic (Isochronous and Interrupt) channels.		
In Non-Scatter/Gather mode, HCTSIZn is defined as follows:			
31	Do Ping (DoPng) This bit is used only for OUT transfers. Setting this field to 1 directs the host to do PING protocol. **Note:** Do not set this bit for IN transfers. If this bit is set for IN transfers it disables the	1'b0	R_W
30:29	PID (Pid) The application programs this field with the type of PID to use for the initial transaction. The host maintains this field for the rest of the transfer. ▪ 2'b00: DATA0 ▪ 2'b01: DATA2 ▪ 2'b10: DATA1 ▪ 2'b11: MDATA (non-control)/SETUP (control)	2'b00	R_W
28:19	Packet Count (PktCnt) This field is programmed by the application with the expected number of packets to be transmitted (OUT) or received (IN). The host decrements this count on every successful transmission or reception of an OUT/IN packet. Once this count reaches zero, the application is interrupted to indicate normal completion.	10'h0	R_W
18:0	Transfer Size (XferSize) For an OUT, this field is the number of data bytes the host sends during the transfer. For an IN, this field is the buffer size that the application has Reserved for the transfer. The application is expected to program this field as an integer multiple of the maximum packet size for IN transactions (periodic and non-periodic).	19'h0	R_W

25.5.4.13 Host Channel-n DMA Address Register (HCDMAn)

✧ Channel_number: 0 ≤n ≤15

✧ Offset: 514h + (Channel_number * 20h)

This register is used by the OTG host in the internal DMA mode to maintain the current buffer pointer

for IN/OUT transactions. The starting DMA address must be DWORD-aligned.

Table 25-41 Host Channel-n DMA Address Register: HCDMAn

<table>
<tr><th>Field</th><th>Description</th><th>Reset</th><th>Access</th></tr>
<tr><td colspan="4">Buffer DMA Mode</td></tr>
<tr><td>31:0</td><td>DMA Address (DMAAddr)
This field holds the start address in the external memory from which the data for the endpoint must be fetched or to which it must be stored. This register is incremented on every AHB transaction.</td><td>"X" if not program med as the register is in SPRAM</td><td>R_W</td></tr>
<tr><td colspan="4">Descriptor DMA Mode</td></tr>
<tr><td colspan="4">Note: For Scatter/Gather DMA mode, this address is the start of the page address where the descriptor list is located.</td></tr>
<tr><td>31:N (Isoc)

31:9 (Non Isoc)</td><td>DMA Address (DMAAddr)
Non-Isochronous:
This field holds the start address of the 512 bytes page. The first descriptor in the list should be located in this address. The first descriptor may be or may not be ready. The core starts processing the list from the CTD value.
Isochronous:
This field holds the address of the 2*(nTD+1) bytes of locations in which the isochronous descriptors are present where N is based on nTD as per Table below
<table>
<tr><td>31:N</td><td>N-1:3</td><td>2:0</td></tr>
<tr><td>Base Address</td><td>Offset</td><td>000</td></tr>
</table>
<table>
<tr><th colspan="2">HS ISOC</th></tr>
<tr><td>nTD</td><td>N</td></tr>
<tr><td>7</td><td>6</td></tr>
<tr><td>15</td><td>7</td></tr>
<tr><td>31</td><td>8</td></tr>
<tr><td>63</td><td>9</td></tr>
<tr><td>127</td><td>10</td></tr>
<tr><td>255</td><td>11</td></tr>
</table>
<table>
<tr><th colspan="2">FS ISOC</th></tr>
<tr><td>nTD</td><td>N</td></tr>
<tr><td>1</td><td>4</td></tr>
<tr><td>3</td><td>5</td></tr>
<tr><td>7</td><td>6</td></tr>
<tr><td>15</td><td>7</td></tr>
<tr><td>31</td><td>8</td></tr>
<tr><td>63</td><td>9</td></tr>
</table></td><td>23'h0</td><td>R_W</td></tr>
<tr><td>N-1:3 (Isoc)

8:3 (Non Isoc)</td><td>Current Transfer Desc (CTD):
Non Isochronous:
This value is in terms of number of descriptors. The values can be from 0 to 63.
0 - 1 descriptor.
63- 64 descriptors.
This field indicates the current descriptor processed in the list.</td><td>6'h0</td><td>R/W</td></tr>
</table>

	This field is updated both by application and the core. For example, if the application enables the channel after programming CTD=5, then the core will start processing the 6th descriptor. The address is obtained by adding a value of (8bytes*5=) 40(decimal) to DMAAddr. **Isochronous:** CTD for isochronous is based on the current frame/μframe value. Need to be set to zero by application.		
2:0	Reserved	3'h0	RO

25.5.4.14 Host Channel-n DMA Buffer Address Register (HCDMABn)

✧ Channel_number: 0 n 15

✧ Offset: 51Ch + (Channel_number * 20h)

This register is present only in case of Scatter/Gather DMA. It is implemented in RAM instead of flop-based implementation. This register holds the current buffer address.

Table 25-42 Host Channel-n DMA Buffer Address Register: HCDMABn

Field	Description	Reset	Access
31:0	Holds the current buffer address. This register is updated as and when the data transfer for the corresponding end point is in progress. This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.	"X" if not program med as the registeris in SPRAM	RO

25.5.4.15 Host Frame List Base Address Register (HFLBAddr)

✧ Offset: 41Ch

This register is present only in case of Scatter/Gather DMA. It is implemented as flops. This register holds the starting address of the Frame list information.

Table 25-43 Host Frame List Base Address Register: HFLBAddr

Field	Description	Reset	Access
31:0	The starting address of the Frame list. This register is used only for Isochronous and Interrupt Channels.	32'h0	R/W

25.5.5 Device Mode Registers

These registers are visible only in Device mode and must not be accessed in Host mode, as the results are unknown. Some of them affect all the endpoints uniformly, while others affect only a specific endpoint.

Device Mode registers fall into two categories:

Device Logical IN Endpoint-Specific Registers

It's instantiates one set of endpoint registers per logical endpoint. A logical endpoint is unidirectional: it can be either IN or OUT. To represent a bidirectional endpoint, two logical endpoints are required, one for the IN direction and the other for the OUT direction. This is also true for control endpoints.
The registers and register fields described in this section can pertain to IN or OUT endpoints, or both, or specific endpoint types as noted.

25.5.5.1 Device Configuration Register (DCFG)

✧ Offset: 800h

This register configures the core in Device mode after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.

Table 25-44 Device Configuration Register: DCFG

Field	Description	Reset	Access
31:26	Resume Validation Period (ResValid) This field controls the period when the core resumes from a suspend. When this bit is set, the core counts for the ResValid number of clock cycles to detect a valid resume. This field is effective only when DCFG.Ena32KHzSusp is set.	6'd2	R_W
25:24	Periodic Scheduling Interval (PerSchIntvl) PerSchIntvl must be programmed only for Scatter/Gather DMA mode. **Description**: This field specifies the amount of time the Internal DMA engine must allocate for fetching periodic IN endpoint data. Based on the number of periodic endpoints, this value must be specified as 25,50 or 75% of (micro)frame. ▪ When any periodic endpoints are active, the internal DMA engine allocates the specified amount of time in fetching periodic IN endpoint data. ▪ When no periodic endpoints are active, then the internal DMA engine services non- periodic endpoints, ignoring this field. ▪ After the specified time within a (micro)frame, the DMA switches to fetching for non- periodic endpoints. ▪ 2'b00: 25% of (micro)frame. ▪ 2'b01: 50% of (micro)frame. ▪ 2'b10: 75% of (micro)frame. ▪ 2'b11: Reserved.	2'b00	R_W
23	Enable Scatter/Gather DMA in Device mode (DescDMA). When the Scatter/Gather DMA option selected during configuration of the RTL, the application can set this bit during initialization to enable the Scatter/Gather DMA operation.	1'b0	R_W

	NOTE: This bit must be modified only once after a reset. The following combinations are available for programming: ▪ GAHBCFG.DMAEn=0,DCFG.DescDMA=0 => Slave mode ▪ GAHBCFG.DMAEn=0,DCFG.DescDMA=1 => Invalid ▪ GAHBCFG.DMAEn=1,DCFG.DescDMA=0 => Buffered DMA mode ▪ GAHBCFG.DMAEn=1,DCFG.DescDMA=1 => Scatter/Gather DMA mode		
22:18	IN Endpoint Mismatch Count (EPMisCnt) This field is valid only in shared FIFO operation. The application programs this filed with a count that determines when the core generates an Endpoint Mismatch interrupt (GINTSTS.EPMis). The core loads this value into an internal counter and decrements it. The counter is reloaded whenever there is a match or when the counter expires. The width of this counter depends on the depth of the Token Queue.	5'h8	R_W
17:13	Reserved		RO
13	Enable Device OUT NAK (EnDevOutNak) This bit enables setting NAK for Bulk OUT endpoints after the transfer is completed when the core is operating in Device Descriptor DMA mode. ▪ 1'b0: The core does not set NAK after Bulk OUT transfer complete ▪ 1'b1: The core sets NAK after Bulk OUT transfer complete This bit is valid only when OTG_EN_DESC_DMA == 1'b1	1'b0	R_W
12:11	Periodic Frame Interval (PerFrInt) Indicates the time within a (micro)frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine if all the isochronous traffic for that (micro)frame is complete. ▪ 2'b00: 80% of the (micro)frame interval ▪ 2'b01: 85% ▪ 2'b10: 90% ▪ 2'b11: 95%	2'h0	R_W
10:4	Device Address (DevAddr) The application must program this field after every SetAddress control command.	7'h0	R_W
3	Enable 32-KHz Suspend Mode (Ena32KHzS) When the USB 1.1 Full-Speed Serial Transceiver Interface is chosen and this bit is set, the core expects the 48-MHz PHY clock to be switched to 32 KHz during a suspend. This bit can only be set if USB 1.1 Full-Speed Serial Transceiver Interface has been selected.	1'd0	R_W

	If USB 1.1 Full-Speed Serial Transceiver Interface has not been selected, this bit must be zero.		
2	Non-Zero-Length Status OUT Handshake (NZStsOUTHShk) The application can use this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage. ▪ 1'b1: Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. ▪ 1'b0: Send the received OUT packet to the application (zero-length or nonzero- length) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register.	1'b0	R_W
1:0	Device Speed (DevSpd) Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. ▪ 2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) ▪ 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) ▪ 2'b10: Low speed (USB 1.1 FS transceiver clock is 48 MHz) ▪ 2'b11: Full speed (USB 1.1 FS transceiver clock is 48 MHz)	2'b0	R_W

25.5.5.2 Device Control Register (DCTL)

✧ Offset: 804h

Table 25-45 device Control Register: DCTL

Field	Description	Reset	Access
31:17	Reserved		RO
17	Enable Continue on BNA (EnContOnBNA) This bit enables the DWC_otg core to continue on BNA for Bulk OUT endpoints. With this feature enabled, when a Bulk OUT endpoint receives a BNA interrupt the core starts processing the descriptor that caused the BNA interrupt after the endpoint re-enables the endpoint. ▪ 1'b0: After receiving BNA interrupt, the core disables the endpoint. When the endpoint is re-enabled by the application, the core starts processing from the DOEPDMA descriptor.	1'b0	R_W

	▪ 1'b1: After receiving BNA interrupt, the core disables the endpoint. When the endpoint is re-enabled by the application, the core starts processing from the descriptor that received the BNA interrupt. This bit is valid only when OTG_EN_DESC_DMA == 1'b1. It is a one-time programmable after reset bit like any other DCTL register bits.		
16	Set NAK automatically on babble (NakOnBble). The core sets NAK automatically for the endpoint on which babble is received.	1'b0	R_W
15	Ignore frame number for isochronous endpoints (IgnrFrmNum) Slave Mode (GAHBCFG.DMAEn=0): This bit is not valid in Slave mode and should not be programmed to 1.Non- Scatter/Gather DMA mode (GAHBCFG.DMAEn=1,DCFG.DescDMA=0): This bit is not used when Threshold mode is enabled and should not be programmed to 1. In non-Scatter/Gather DMA mode, the application receives transfer complete interrupt after transfers for multiple (micro)frames are completed. ▪ When Scatter/Gather DMA mode is disabled, this field is used by the application to enable periodic transfer interrupt. The application can program periodic endpoint transfers for multiple (micro)frames. – 0: Periodic transfer interrupt feature is disabled; the application must program transfers for periodic endpoints every (micro)frame – 1: Packets are not flushed when an ISOC IN token is received for an elapsed frame. The core ignores the frame number, sending packets as soon as the packets are ready, and the corresponding token is received. This field is also used by the application to enable periodic transfer interrupts. Scatter/Gather DMA Mode (GAHBCFG.DMAEn=1,DCFG.DescDMA=1): This bit is not applicable to high-speed, high-bandwidth transfers and should not be programmed to 1. In addition, this bit is not used when Threshold mode is enabled and should not be programmed to 1. ▪ 0: The core transmits the packets only in the frame number in which they are intended to be transmitted. – 1: Packets are not flushed when an ISOC IN token is received for an elapsed frame. The core ignores the frame number, sending packets as soon as the	1'b0	R_W

	packets are ready, and the corresponding token is received. When this bit is set, there must be only one packet per descriptor.		
14:13	Global Multi Count (GMC) GMC must be programmed only once after initialization. Applicable only for Scatter/Gather DMA mode. This indicates the number of packets to be serviced for that end point before moving to the next end point. It is only for non- periodic end points. ▪ 2'b00: Invalid. ▪ 2'b01: 1 packet. ▪ 2'b10: 2 packets. ▪ 2'b11: 3 packets. The value of this field automatically changes to 2'h1 when DCFG.DescDMA is set to 1. When Scatter/Gather DMA mode is disabled, this field is reserved and reads 2'b00.	2'h0	R_W
12	Reserved		RO
11	Power-On Programming Done (PWROnPrgDone) The application uses this bit to indicate that register programming is completed after a wake-up from Power Down mode.	1'b0	WO
10	Clear Global OUT NAK (CGOUTNak) A write to this field clears the Global OUT NAK.	1'b0	WO
9	Set Global OUT NAK (SGOUTNak) A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints. The application must set the this bit only after making sure that the Global OUT NAK Effective bit in the Core Interrupt Register (GINTSTS.GOUTNakEff) is cleared.	1'b0	WO
8	Clear Global Non-periodic IN NAK (CGNPInNak) A write to this field clears the Global Non-periodic IN NAK.	1'b0	WO
7	Set Global Non-periodic IN NAK (SGNPInNak) A write to this field sets the Global Non-periodic IN NAK.The application uses this bit to send a NAK handshake on all non-periodic IN endpoints. The core can also set this bit when a timeout condition is detected on a non-periodic endpoint in shared FIFO operation. The application must set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register (GINTSTS.GINNakEff) is cleared.	1'b0	WO

6:4	Test Control (TstCtl) ▪ 3'b000: Test mode disabled ▪ 3'b001: Test_J mode ▪ 3'b010: Test_K mode ▪ 3'b011: Test_SE0_NAK mode ▪ 3'b100: Test_Packet mode ▪ 3'b101: Test_Force_Enable Others: Reserved	3'h0	R_W
3	Global OUT NAK Status (GOUTNakSts) ▪ 1'b0: A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. ▪ 1'b1: No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped.	1'b0	WO
2	Global Non-periodic IN NAK Status (GNPINNakSts) ▪ 1'b0: A handshake is sent out based on the data availability in the transmit FIFO. ▪ 1'b1: A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.	1'b0	WO
1	Soft Disconnect (SftDiscon) The application uses this bit to signal the DWC_otg core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device does not receive signals on the USB. The core stays in the disconnected state until the application clears this bit. The minimum duration for which the core must keep this bit set is specified in Table 25-53. ▪ 1'b0: Normal operation. When this bit is cleared after a soft disconnect, the core drives the phy_opmode_o signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. When the device is reconnected, the USB host restarts device enumeration. ▪ 1'b1: The core drives the phy_opmode_o signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host. Note: This bit can be also used for ULPI/FS Serial interfaces. Note: This bit is not impacted by a soft reset.	1'b1	R_W
0	Remote Wakeup Signaling (RmtWkUpSig) When the application sets this bit, the core initiates remote signaling to wake the USB host. The application must set this bit to instruct the core to exit the Suspend state. As specified in the	1'b0	R_W

	USB 2.0 specification, the application must clear this bit 1-15 ms after setting it.If LPM is enabled and the core is in the L1 (Sleep) state, when the application sets this bit, the core initiates L1 remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Sleep state. As specified in the LPM specification, the hardware automatically clears this bit 50 µs ($T_{L1DevDrvResume}$) after being set by the application. The application must not set this bit when GLPMCFG bRemoteWake from the previous LPM transaction is zero.		

Table 25-46 lists the minimum duration under various conditions for which the Soft Disconnect (SftDiscon) bit must be set for the USB host to detect a device disconnect. To accommodate clock jitter, it is recommended that the application add some extra delay to the specified minimum duration.

Table 25-46 Minimum Duration for Soft Disconnect

Operating Speed	Device State	Minimum Duration
High speed	Suspended	1 ms + 2.5 µs
High speed	Idle	3 ms + 2.5 µs
High speed	Not Idle or Suspended (Performing transactions)	125 µs
Full speed/Low speed	Suspended	1 ms + 2.5 µs
Full speed/Low speed	Idle	2.5 µs
Full speed/Low speed	Not Idle or Suspended (Performing transactions)	2.5 µs

25.5.5.3 Device Status Register (DSTS)

✧ Offset: 808h

This register indicates the status of the core with respect to USB-related events. It must be read on interrupts from Device All Interrupts (DAINT) register.

Table 25-47 Device Status Register: DSTS

Field	Description	Reset	Access
31:24	Reserved		RO
23:22	Device Line Status (DevLnSts) Indicates the current logic level USB data lines ■ Bit [23]: Logic level of D+ ■ Bit [22]: Logic level of D–	2'b00	RO
21:8	Frame or Microframe Number of the Received SOF (SOFFN) When the core is operating at high speed, this field contains a microframe number. When the core is operating at full or low speed, this field contains	14'h0	RO

	a frame number. **Note:** This register may return a non zero value if read immediately after power on reset. In case the register bit reads non zero immediately after power on reset it does not indicate that SOF has been received from the host. The read value of this interrupt is valid only after a valid connection between host and device is established.		
7:4	Reserved		RO
3	Erratic Error (ErrticErr) The core sets this bit to report any erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted for at least 2 ms, due to PHY error) seen on the UTMI+. Due to erratic errors, the DWC_otg core goes into Suspended state and an interrupt is generated to the application with Early Suspend bit of the Core Interrupt register (GINTSTS.ErlySusp). If the early suspend is asserted due to an erratic error, the application can only perform a soft disconnect recover.	1'b0	RO
2:1	Enumerated Speed (EnumSpd) Indicates the speed at which the DWC_otg core has come up after speed detection through a chirp sequence. ▪ 2'b00: High speed (PHY clock is running at 30 or 60 MHz) ▪ 2'b01: Full speed (PHY clock is running at 30 or 60 MHz) ▪ 2'b10: Low speed (PHY clock is running at 48 MHz, internal phy_clk at 6 MHz) ▪ 2'b11: Full speed (PHY clock is running at 48 MHz) Low speed is not supported for devices using a UTMI+ PHY.	2'h01	RO
0	Suspend Status (SuspSts) In Device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspended state when there is no activity on the utmi_linestate signal for an extended period of time. The core comes out of the suspend: ▪ When there is any activity on the utmi_linestate signal ▪ When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig).	1'b0	RO

25.5.5.4 Device IN Endpoint Common Interrupt Mask Register (DIEPMSK)

✧ Offset: 810h

This register works with each of the Device IN Endpoint Interrupt (DIEPINTn) registers for all endpoints to generate an interrupt per IN endpoint. The IN endpoint interrupt for a specific status in the DIEPINTn register can be masked by writing to the corresponding bit in this register. Status bits are

masked by default.

✧ Mask interrupt: 1'b0

✧ Unmask interrupt: 1'b1

Table 25-48 Device IN Endpoint Common Interrupt Mask Register: DIEPMSK

Field	Description	Reset	Access
31:14	Reserved		RO
13	NAK interrupt Mask (NAKMsk)	1'h0	R_W
12:10	Reserved		RO
9	BNA Interrupt Mask (BNAInIntrMsk) This bit is valid only when Device Descriptor DMA is enabled.	1'b0	R_W
8	Fifo Underrun Mask (TxfifoUndrnMsk)	1'b0	R_W
7	Reserved		RO
6	IN Endpoint NAK Effective Mask (INEPNakEffMsk)	1'b0	R_W
5	IN Token received with EP Mismatch Mask (INTknEPMisMsk)	1'b0	R_W
4	IN Token Received When TxFIFO Empty Mask (INTknTXFEmpMsk)	1'b0	R_W
3	Timeout Condition Mask (TimeOUTMsk) (Non-isochronous endpoints)	1'b0	R_W
2	AHB Error Mask (AHBErrMsk)	1'b0	R_W
1	Endpoint Disabled Interrupt Mask (EPDisbldMsk)	1'b0	R_W
0	Transfer Completed Interrupt Mask (XferComplMsk)	1'b0	R_W

25.5.5.5 Device OUT Endpoint Common Interrupt Mask Register (DOEPMSK)

✧ Offset: 814h

This register works with each of the Device OUT Endpoint Interrupt (DOEPINTn) registers for all endpoints to generate an interrupt per OUT endpoint. The OUT endpoint interrupt for a specific status in the DOEPINTn register can be masked by writing into the corresponding bit in this register. Status bits are masked by default.

✧ Mask interrupt: 1'b0

✧ Unmask interrupt: 1'b1

Table 25-49 Device OUT Endpoint Common Interrupt Mask Register: DOEPMSK

Field	Description	Reset	Access
31:15	Reserved		RO
14	NYET Interrupt Mask (NYETMsk)	1'h0	R_W
13	NAK Interrupt Mask (NAKMsk)	1'h0	R_W
12	Babble Interrupt Mask (BbleErrMsk)	1'h0	R_W
11:10	Reserved		RO
9	BNA interrupt Mask (BnaOutIntrMsk)	1'h0	R_W

8	OUT Packet Error Mask (OutPktErrMsk)	1'b0	R_W
7	Reserved		RO
6	Back-to-Back SETUP Packets Received Mask (Back2BackSETup) Applies to control OUT endpoints only.	1'b0	R_W
5	Reserved		RO
4	OUT Token Received when Endpoint Disabled Mask (OUTTknEPdisMsk) Applies to control OUT endpoints only.	1'b0	R_W
3	SETUP Phase Done Mask (SetUPMsk) Applies to control endpoints only.	1'b0	R_W
2	AHB Error (AHBErrMsk)	1'b0	R_W
1	Endpoint Disabled Interrupt Mask (EPDisbldMsk)	1'b0	R_W
0	Transfer Completed Interrupt Mask (XferComplMsk)	1'b0	R_W

25.5.5.6 Device All Endpoints Interrupt Register (DAINT)

✧ Offset: 818h

When a significant event occurs on an endpoint, a Device All Endpoints Interrupt register interrupts the application using the Device OUT Endpoints Interrupt bit or Device IN Endpoints Interrupt bit of the Core Interrupt register (GINTSTS.OEPInt or GINTSTS.IEPInt, respectively). There is one interrupt bit per endpoint, up to a maximum of 16 bits for OUT endpoints and 16 bits for IN endpoints.

For a bidirectional endpoint, the corresponding IN and OUT interrupt bits are used. Bits in this register are set and cleared when the application sets and clears bits in the corresponding Device Endpoint-n Interrupt register (DIEPINTn/DOEPINTn).

Table 25-50 Device All Endpoints Interrupt Register: DAINT

Field	Description	Reset	Access
31:16	OUT Endpoint Interrupt Bits (OutEPInt) One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15	16'h0	RO
15:0	IN Endpoint Interrupt Bits (InEpInt) One bit per IN Endpoint: Bit 0 for IN endpoint 0, bit 15 for endpoint 15	16'h0	RO

25.5.5.7 Device All Endpoints Interrupt Mask Register (DAINTMSK)

✧ Offset: 81Ch

The Device Endpoint Interrupt Mask register works with the Device Endpoint Interrupt register to interrupt the application when an event occurs on a device endpoint. However, the Device All Endpoints Interrupt (DAINT) register bit corresponding to that interrupt is still set.

✧ Mask Interrupt: 1'b0

✧ Unmask Interrupt: 1'b1

Table 25-51 Device Endpoints Interrupt Mask Register: DAINTMSK

Field	Description	Reset	Access
31:16	OUT EP Interrupt Mask Bits (OutEpMsk) One per OUT Endpoint: Bit 16 for OUT EP 0, bit 31 for OUT EP 15 The value of this field depends on the number of OUT endpoints that are configured.	16'h0	R_W
15:0	IN EP Interrupt Mask Bits (InEpMsk) One bit per IN Endpoint: Bit 0 for IN EP 0, bit 15 for IN EP 15 The value of this field depends on the number of IN endpoints that are configured.	16'h0	R_W

25.5.5.8 Device IN Token Sequence Learning Queue Read Register 1 (DTKNQR1)

✧ Offset: 820h

This register is valid only in non-periodic Shared FIFO operation (OTG_EN_DED_TX_FIFO = 0). The queue is 4 bits wide to store the endpoint number. A read from this register returns the first 5 endpoint entries of the IN Token Sequence Learning Queue. When the queue is full, the new token is pushed into the queue and oldest token is discarded.

Table 25-52 Device IN Token Sequence Learning Queue Read Register 1: DTKNQR1

Field	Description	Reset	Access
31:8	Endpoint Token (EPTkn) Four bits per token represent the endpoint number of the token: ▪ Bits [31:28]: Endpoint number of Token 5 ▪ Bits [27:24]: Endpoint number of Token 4 ..... ▪ Bits [15:12]: Endpoint number of Token 1 ▪ Bits [11:8]: Endpoint number of Token 0	24'h0	RO
7	Wrap Bit (WrapBit) This bit is set when the write pointer wraps. It is cleared when the learning queue is cleared.	1'b0	RO
6:5	Reserved		RO
4:0	IN Token Queue Write Pointer (INTknWPtr)	5'h0	RO

25.5.5.9 Device IN Token Sequence Learning Queue Read Register 2 (DTKNQR2)

✧ Offset: 0824h

This register is valid only in shared non-periodic Shared FIFO operation (OTG_EN_DED_TX_FIFO = 0).

A read from this register returns the next 8 endpoint entries of the learning queue.

Table 25-53 Device IN Token Sequence Learning Queue Register 2: DTKNQR2

Field	Description	Reset	Access
31:0	Endpoint Token (EPTkn) Four bits per token represent the endpoint number of the token: ▪ Bits [31:28]: Endpoint number of Token 13 ▪ Bits [27:24]: Endpoint number of Token 12 ..... ▪ Bits [7:4]: Endpoint number of Token 7 ▪ Bits [3:0]: Endpoint number of Token 6	32'h0	RO

25.5.5.10 Device IN Token Sequence Learning Queue Read Register 3 (DTKNQR3)

✧ Offset: 0830h

This register is valid only in non-periodic Shared FIFO operation (OTG_EN_DED_TX_FIFO = 0).

A read from this register returns the next 8 endpoint entries of the learning queue.

Table 25-54 Device IN Token Sequence Learning Queue Register 3: DTKNQR3

Field	Description	Reset	Access
31:0	Endpoint Token (EPTkn) Four bits per token represent the endpoint number of the token: ▪ Bits [31:28]: Endpoint number of Token 21 ▪ Bits [27:24]: Endpoint number of Token 20 ..... ▪ Bits [7:4]: Endpoint number of Token 15 ▪ Bits [3:0]: Endpoint number of Token 14	32'h0	RO

25.5.5.11 Device IN Token Sequence Learning Queue Read Register 4 (DTKNQR4)

✧ Offset: 0834h

This register is valid only in non-periodic Shared FIFO operation (OTG_EN_DED_TX_FIFO = 0).
A read from this register returns the last 8 endpoint entries of the learning queue.

Table 25-55 Device IN Token Sequence Learning Queue Register 4: DTKNQR4

Field	Description	Reset	Access
31:0	Endpoint Token (EPTkn)	32'h0	RO

	Four bits per token represent the endpoint number of the token: ▪ Bits [31:28]: Endpoint number of Token 29 ▪ Bits [27:24]: Endpoint number of Token 28 ..... ▪ Bits [7:4]: Endpoint number of Token 23 ▪ Bits [3:0]: Endpoint number of Token 22		

25.5.5.12 Device VBUS Discharge Time Register (DVBUSDIS)

✧ Offset: 0828h

This register specifies the VBUS discharge time after VBUS pulsing during SRP.

Table 25-56 Device VBUS Discharge Time Register: DVBUSDIS

Field	Description	Reset	Acces
31:16	Reserved		RO
15:0	Device VBUS Discharge Time (DVBUSDis) Specifies the VBUS discharge time after VBUS pulsing during SRP. This value equals: VBUS discharge time in PHY clocks / 1,024 The value you use depends whether the PHY is operating at 30 MHz (16-bit data width) or 60 MHz (8-bit data width). Depending on your VBUS load, this value can need adjustment.	30 MHz: 16'h0B8F 60 MHz: 16'h17D7	R_W

25.5.5.13 Device VBUS Pulsing Time Register (DVBUSPULSE)

✧ Offset: 082Ch

This register specifies the VBUS pulsing time during SRP.

Table 25-57 Device VBUS Pulsing Time Register (DVBUSPULSE)

Field	Description	Reset	Acces
31:12	Reserved		RO
11:0	Device VBUS Pulsing Time (DVBUSPulse) Specifies the VBUS pulsing time during SRP. This value equals: VBUS pulsing time in PHY clocks / 1,024 The value you use depends whether the PHY is operating at 30 MHz (16-bit data width) or 60 MHz (8-bit data width).	30 MHz: 12'h2C6 60 MHz: 12'h5B8	R_W

25.5.5.14 Device Threshold Control Register (DTHRCTL)

✧ Offset: 830h

This register is valid only for device mode in Dedicated FIFO operation (OTG_EN_DED_TX_FIFO=1). Thresholding is not supported in Slave mode and so this register must not be programmed in Slave mode.

For threshold support, the AHB must be run at 60 MHz or higher.

Table 25-58 Device Threshold Control Register (DTHRCTL)

Field	Description	Reset	Acces
31:28	Reserved		RO
27	Arbiter Parking Enable (ArbPrkEn) This bit controls internal DMA arbiter parking for IN endpoints. When thresholding is enabled and this bit is set to one, then the arbiter parks on the IN endpoint for which there is a token received on the USB. This is done to avoid getting into underrun conditions. By default the parking is enabled.	1'b1	R_W
26	Reserved		RO
25:17	Receive Threshold Length (RxThrLen) This field specifies Receive thresholding size in DWORDS. This field also specifies the amount of data received on the USB before the core can start transmitting on the AHB. The threshold length has to be at least eight DWORDS. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).	9'h8	R_W
16	Receive Threshold Enable (RxThrEn) When this bit is set, the core enables thresholding in the receive direction. Note: Synopsys recommends that you do not enable RxThrEn, because it may cause issues in the RxFIFO especially during error conditions such as RxError and Babble.	1'b0	R_W
15:13	Reserved		RO
12:11	AHB Threshold Ratio (AHBThrRatio) These bits define the ratio between the AHB threshold and the MAC threshold for the transmit path only. The AHB threshold always remains less than or equal to the USB threshold, because this does not increase overhead. Both the AHB and the MAC threshold must be DWORD-aligned. The application needs to program TxThrLen and the AHBThrRatio to make the AHB Threshold value DWORD aligned. If the AHB threshold value is not DWORD aligned, the core might not behave correctly. When programming the TxThrLen and AHBThrRatio, the application must ensure that the minimum AHB threshold value does not go below 8 DWORDS to meet the USB turnaround time requirements.	2'h0	R_W
10:2	Transmit Threshold Length (TxThrLen) This field specifies Transmit thresholding size in DWORDS. This field also forms the MAC threshold and specifies the amount of data, in bytes, to be in the corresponding endpoint transmit FIFO before the core can start a transmit on the USB. When the value of AHBThrRatio is 2'h00, the threshold length must be at least 8 DWORDS. If the AHBThrRatio is nonzero, the application must ensure that	9'h8	R_W

	the AHB threshold value does not go below the recommended 8 DWORDs. This field controls both isochronous and non-isochronous IN endpoint thresholds. The recommended value for ThrLen is to be the same as the programmed AHB Burst Length (GAHBCFG.HBstLen).		
1	ISO IN Endpoints Threshold Enable. (ISOThrEn) When this bit is set, the core enables thresholding for isochronous IN endpoints.	1'b0	R_W
0	Non-ISO IN Endpoints Threshold Enable. (NonISOThrEn) When this bit is set, the core enables thresholding for Non Isochronous IN endpoints.	1'b0	R_W

25.5.5.15 Device IN Endpoint FIFO Empty Interrupt Mask Register: (DIEPEMPMSK)

✧ Offset: 834 h

This register is valid only in Dedicated FIFO operation (OTG_EN_DED_TX_FIFO = 1).

This register is used to control the IN endpoint FIFO empty interrupt generation (DIEPINTn.TxfEmp).

✧ Mask interrupt: 1'b0

✧ Unmask interrupt: 1'b1

Table 25-59 Device IN Endpoint FIFO Empty Interrupt Mask Register: DIEPEMPMSK

Field	Description	Reset	Acces
31:16	Reserved		RO
15:0	IN EP Tx FIFO Empty Interrupt Mask Bits (InEpTxfEmpMsk) These bits acts as mask bits for DIEPINTn. TxFEmp interrupt One bit per IN Endpoint: ▪ Bit 0 for IN endpoint 0 ... ▪ Bit 15 for endpoint 15	16'h0	R_W

25.5.5.16 Device Each Endpoint Interrupt Register (DEACHINT)

✧ Offset: 838h

Dependency: This register is available in device mode and only when parameter "Name: OTG_MULTI_PROC_INTRPT".

There is one interrupt bit per endpoint, up to a maximum of 16 bits for OUT endpoints and 16 bits for IN endpoints. For a bidirectional endpoint, the corresponding IN and OUT interrupt bits are used.

✧ Bits in this register are set and cleared when the application sets and clears bits in the corresponding Device Endpoint-n Interrupt register (DIEPINTn/DOEPINTn).

✧ The interrupt is automatically cleared once the DOEPINTn / DIEPINTn interrupt is cleared by the application.

Table 25-60 Device Each Endpoint Interrupt Register: DEACHINT

Field	Description	Reset	Acces
31:16	OUT Endpoint Interrupt Bits (EchOutEPInt) One bit per OUT endpoint: ▪ Bit 16 for OUT endpoint 0 ... ▪ Bit 31 for OUT endpoint 15	16'h0	RO
15:0	IN Endpoint Interrupt Bits (EchInEpInt) One bit per IN Endpoint: ▪ Bit 0 for IN endpoint 0 ... ▪ Bit 15 for IN endpoint 15	16'h0	RO

25.5.5.17 Device Each Endpoint Interrupt Register Mask (DEACHINTMSK)

✧ Offset: 83Ch

Dependency: This register is available only when parameter "Name: OTG_MULTI_PROC_INTRPT".

The Device Each Endpoint Interrupt Mask register works with the Device Each Endpoint Interrupt register to interrupt the application when an event occurs on a device endpoint. However, the Device Each Endpoints Interrupt (DEACHINT) register bit corresponding to that interrupt remains set.

✧ Mask Interrupt: 1.b0
✧ Unmask Interrupt: 1.b1

Table 25-61 Device Each Endpoint Interrupt Register Mask: DEACHINTMSK

Field	Description	Reset	Acces
31:16	OUT EP Interrupt Mask Bits (EchOutEpMsk) One per OUT Endpoint: ▪ Bit 16 for IN endpoint 0 ... ▪ Bit 31 for IN endpoint 15	16'h0	R_W
15:0	IN EP Interrupt Mask Bits (EchInEpMsk) One bit per IN Endpoint: ▪ Bit 0 for IN endpoint 0 ... ▪ Bit 15 for IN endpoint 15	16'h0	R_W

25.5.5.18 Device Each In Endpoint-*n* Interrupt Register (DIEPEACHMSKn)

✧ Offset 840h
✧ Endpoint_number: 0=< n =< 15
✧ Offset for IN endpoints: 840h + (Endpoint_number * 4h)

Dependency: This register is available in device mode and only when parameter "Name: OTG_MULTI_PROC_INTRPT".

These registers are endpoint-specific mask registers for (DIEPINTn). The IN endpoint interrupt for a specific status in the DIEPINTn register can be masked by writing 0 to the corresponding bit in this register. Status bits are masked by default.

✧ Mask interrupt: 1'b0

✧ Unmask interrupt: 1'b1

Table 25-62 Device Each In Endpoint-n Interrupt Register: DIEPEACHMSKn

Field	Description	Reset	Acces
31:14	Reserved		RO
13	NAK interrupt Mask (NAKMsk)	1'b0	R_W
12:10	Reserved		RO
9	BNA interrupt Mask (BNAInIntrMsk)	1'b0	R_W
8	Fifo Under run Mask (TxfifoUndrnMsk)	1'b0	R_W
7	Reserved		RO
6	IN Endpoint NAK Effective Mask (INEPNakEffMsk)	1'b0	R_W
5	IN Token received with EP Mismatch Mask (INTknEPMisMsk)	1'b0	R_W
4	IN Token Received When TxFIFO Empty Mask (INTknTXFEmpMsk)	1'b0	R_W
3	Timeout Condition Mask (TimeOUTMsk) (Non-isochronous endpoints)	1'b0	R_W
2	AHB Error Mask (AHBErrMsk)	1'b0	R_W
1	Endpoint Disabled Interrupt Mask (EPDisbldMsk)	1'b0	R_W
0	Transfer Completed Interrupt Mask (XferComplMsk)	1'b0	R_W

25.5.5.19 Device Each Out Endpoint-*n* Interrupt Register (DOEPEACHMSKn)

✧ Offset 880h

✧ Endpoint_number: 0 =< n =< 15

✧ Offset for OUT endpoints: 880h + (Endpoint_number * 4h)

Dependency: This register is available in device mode and only when parameter OTG_MULTI_PROC_INTRPT=1.

These registers are endpoint specific mask registers for (DOEPINTn). The OUT endpoint interrupt for a specific status in the DOEPINTn register can be masked by writing 0 to the corresponding bit in this register. Status bits are masked by default.

✧ Mask interrupt: 1'b0

✧ Unmask interrupt: 1'b1

Table 25-63 Device Each Out Endpoint-n Interrupt Register: DOEPEACHMSKn

Field	Description	Reset	Acces
31:15	Reserved		RO
14	NYET interrupt Mask (NYETMsk)	1'b0	R_W
13	NAK interrupt Mask (NAKMsk)	1'b0	R_W
12	Babble interrupt Mask (BbleErrMsk)	1'b0	R_W
11:10	Reserved		RO
9	BNA interrupt Mask (BnaOutIntrMsk)	1'b0	R_W
8	OUT Packet Error Mask (OutPktErrMsk)	1'b0	R_W
7	Reserved		RO
6	Back-to-Back SETUP Packets Received Mask (Back2BackSETup) Applies to control OUT endpoints only.	1'b0	R_W
5	Reserved		RO
4	OUT Token Received when Endpoint Disabled Mask (OUTTknEPdisMsk) Applies to control OUT endpoints only.	1'b0	R_W
3	SETUP Phase Done Mask (SetUPMsk) Applies to control endpoints only.	1'b0	R_W
2	AHB Error (AHBErrMsk)	1'b0	R_W
1	Endpoint Disabled Interrupt Mask (EPDisbldMsk)	1'b0	R_W
0	Transfer Completed Interrupt Mask (XferComplMsk)	1'b0	R_W

25.5.5.20 Device Control IN Endpoint 0 Control Register (DIEPCTL0)

✧ Offset: 900h

This section describes the Control IN Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1-15.

Table 25-64 Device Control IN Endpoint 0 Control Register: DIEPCTL0

Field	Description	Reset	Acces
31	Endpoint Enable (EPEna) ▪ When Scatter/Gather DMA mode is enabled, for IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. ▪ When Scatter/Gather DMA mode is disabled—such as in buffer-pointer based DMA mode—this bit indicates that data is ready to be transmitted on the endpoint. The core clears this bit before setting the following interrupts on this endpoint: ▪ Endpoint Disabled ▪ Transfer Completed	1'b0	R_WS_SC
30	Endpoint Disable (EPDis)	1'b0	R_WS_SC

	The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint. This bit is valid only when DMA mode is enabled.		
29:28	Reserved		RO
27	Set NAK (SNAK) A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint.	1'b0	WO
26	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint.	1'b0	WO
25:22	TxFIFO Number (TxFNum) For Shared FIFO operation, this value is always set to 0, indicating that control IN endpoint 0 data is always written in the Non-Periodic Transmit FIFO. For Dedicated FIFO operation, this value is set to the FIFO number that is assigned to IN Endpoint 0.	4'h0	R_W
21	STALL Handshake (Stall) The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority.	1'b0	R_WS_SC
20	Reserved		RO
19:18	Endpoint Type (EPType) Hardcoded to 00 for control.	2'h0	RO
17	NAK Status (NAKSts) Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status 1'b1: The core is transmitting NAK handshakes on this endpoint. When this bit is set, either by the application or core, the core stops transmitting data, even if there is data available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	1'b0	RO
16	Reserved		RO
15	USB Active Endpoint (USBActEP) This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.	1'b1	RO

14:11	Next Endpoint (NextEp) Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is not set. This field is not valid in Slave mode. **Note**: This field is valid only for Shared FIFO operations.	4'b0	R_W
10:2	Reserved		RO
1:0	Maximum Packet Size (MPS) Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint.	2'h0	R_W

25.5.5.21 Device Control OUT Endpoint 0 Control Register (DOEPCTL0)

✧ Offset: B00h

This section describes the Control OUT Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1-15.

Table 25-65 Device OUT Endpoint 0 Control Register: DOEPCTL0

Field	Description	Reset	Access
31	Endpoint Enable (EPEna) When Scatter/Gather DMA mode is enabled, for OUT endpoints this bit indicates that the descriptor structure and data buffer to receive data is setup. ▪ When Scatter/Gather DMA mode is disabled—(such as for buffer-pointer based DMA mode)—this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: ▪ SETUP Phase Done ▪ Endpoint Disabled ▪ Transfer Completed **Note**: In DMA mode, this bit must be set for the core to transfer SETUP data packets into memory.	1'b0	R_WS_SC
30	Endpoint Disable (EPDis) The application cannot disable control OUT endpoint 0.	1'b0	RO
29:28	Reserved		RO
27	Set NAK (SNAK) A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK	1'b0	WO

	handshakes on an endpoint. The core can also set bit on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.		
26	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint.	1'b0	WO
25:22	Reserved		RO
21	STALL Handshake (Stall) The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	1'b0	R_WS_SC
20	Snoop Mode (Snp) This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.	1'b0	R_W
19:18	Endpoint Type (EPType) Hardcoded to 2'b00 for control.	2'h0	RO
17	NAK Status (NAKSts) Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit, the core stops receiving data, even if there is space in the RxFIFO to accommodate the incoming packet. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	1'b0	RO
16	Reserved		RO
15	USB Active Endpoint (USBActEP) This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces.	1'b1	RO
14:2	Reserved		RO
1:0	Maximum Packet Size (MPS) The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. ▪ 2'b00: 64 bytes ▪ 2'b01: 32 bytes ▪ 2'b10: 16 bytes ▪ 2'b11: 8 bytes	2'h0	RO

25.5.5.22 Device Endpoint-*n* Control Register (DIEPCTLn/DOEPCTLn)

✧ Endpoint_number: 1 ≤n ≤15

✧ Offset for IN endpoints: 900h + (Endpoint_number * 20h)

✧ Offset for OUT endpoints: B00h + (Endpoint_number * 20h)

The application uses this register to control the behavior of each logical endpoint other than endpoint 0.

Table 25-66 Device Endpoint-n Control Register: DIEPCTLn/DOEPCTLn

Field	Description	Reset	Access
31	Endpoint Enable (EPEna) Applies to IN and OUT endpoints. ▪ When Scatter/Gather DMA mode is enabled, ▪ For IN endpoints this bit indicates that the descriptor structure and data buffer with data ready to transmit is setup. ▪ For OUT endpoint it indicates that the descriptor structure and data buffer to ▪ receive data is setup. ▪ When Scatter/Gather DMA mode is enabled—such as for buffer-pointer based DMA mode: – For IN endpoints, this bit indicates that data is ready to be transmitted on the endpoint. – For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB. – The core clears this bit before setting any of the following interrupts on this endpoint. ▪ SETUP Phase Done ▪ Endpoint Disabled ▪ Transfer Completed **Note**: For control endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.	1'b0	R_WS_SC
30	Endpoint Disable (EPDis) Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.	1'b0	R_WS_SC
29	Set DATA1 PID (SetD1PID) Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode.	1'b0	WO WO

	Set Odd (micro)frame (SetOddFr) Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro)frame (EO_FrNum) field to odd (micro)frame. This field is not applicable for Scatter/Gather DMA mode.		
28	Set DATA0 PID (SetD0PID) Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. In non-Scatter/Gather DMA mode: Set Even (micro)frame (SetEvenFr) Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro)frame (EO_FrNum) field to even (micro) frame. When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is in the transmit descriptor structure. The frame in which to receive data is updated in receive descriptor structure.	1'b0	WO
27	Set NAK (SNAK) Applies to IN and OUT endpoints. A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.	1'b0	WO
26	Clear NAK (CNAK) Applies to IN and OUT endpoints. A write to this bit clears the NAK bit for the endpoint.	1'b0	WO
25:22	TxFIFO Number (TxFNum) ***Shared FIFO Operation***—non-periodic endpoints must set this bit to zero. Periodic endpoints must map this to the corresponding Periodic TxFIFO number. ▪ 4'h0: Non-Periodic TxFIFO ▪ Others: Specified Periodic TxFIFO.number **Note**: An interrupt IN endpoint can be configured as a non-periodic endpoint for applications such as mass storage. The core treats an IN endpoint as a non-periodic endpoint if the TxFNum field is set to 0. Otherwise, a separate periodic FIFO must be allocated for an interrupt IN endpoint, and the number of this FIFO must be programmed into the TxFNum field. Configuring an interrupt IN endpoint as a non-periodic	4'h0	R_W

	endpoint saves the extra periodic FIFO area. ***Dedicated FIFO Operation***—these bits specify the FIFO number associated with this endpoint. Each active IN endpoint must be programmed to a separate FIFO number. This field is valid only for IN endpoints.		
21	STALL Handshake (Stall) Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	1'h0	R_W R_WS_SC
20	Snoop Mode (Snp) Applies to OUT endpoints only. This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.	1'b0	R_W
19:18	Endpoint Type (EPType) Applies to IN and OUT endpoints. This is the transfer type supported by this logical endpoint. ▪ 2'b00: Control ▪ 2'b01: Isochronous ▪ 2'b10: Bulk ▪ 2'b11: Interrupt	2'h0	R_W
17	NAK Status (NAKSts) Applies to IN and OUT endpoints. Indicates the following: ▪ 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. ▪ 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO.	1'b0	RO

	Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.		
16	Endpoint Data PID (DPID) Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. The applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. ▪ 1'b0: DATA0 ▪ 1'b1: DATA1 This field is applicable both for Scatter/Gather DMA mode and non-Scatter/Gather DMA mode. Even/Odd (Micro)Frame (EO_FrNum) In non-Scatter/Gather DMA mode: Applies to isochronous IN and OUT endpoints only. Indicates the (micro)frame number in which the core transmits/receives isochronous data for this endpoint. The application must program the even/odd (micro) frame number in which it intends to transmit/receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register. ▪ 1'b0: Even (micro)frame ▪ 1'b1: Odd (micro)frame When Scatter/Gather DMA mode is enabled, this field is reserved. The frame number in which to send data is provided in the transmit descriptor structure. The frame in which data is received is updated in receive descriptor structure.	1'b0	RO
15	USB Active Endpoint (USBActEP) Applies to IN and OUT endpoints. Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.	1'b0	R_W_SC
14:11	Next Endpoint (NextEp) Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is low. This field is not valid in Slave mode operation. **Note**: This field is valid only for Shared FIFO operations.	4'b0	R_W
10:0	Maximum Packet Size (MPS)	11'h0	R_W

	Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current		

25.5.5.23 Device Endpoint-*n* Interrupt Register (DIEPINTn/DOEPINTn)

✧ Endpoint_number: 0 ≤n ≤15
✧ Offset for IN endpoints: 908h + (Endpoint_number * 20h)
✧ Offset for OUT endpoints: B08h + (Endpoint_number * 20h)

This register indicates the status of an endpoint with respect to USB- and AHB-related events. The application must read this register when the OUT Endpoints Interrupt bit or IN Endpoints Interrupt bit of the Core Interrupt register (GINTSTS.OEPInt or GINTSTS.IEPInt, respectively) is set. Before the application can read this register, it must first read the Device All Endpoints Interrupt (DAINT) register to get the exact endpoint number for the Device Endpoint-n Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the DAINT and GINTSTS registers.

Table 25-67 Device Endpoint-n Interrupt Register: DIEPINTn/DOEPINTn

Field	Description	Reset	Access
31:15	Reserved		RO
14	NYET interrupt (NYETIntrpt) The core generates this interrupt when a NYET response is transmitted for a non isochronous OUT endpoint.	1'b0	R_SS_WC
13	NAK interrupt (NAKIntrpt) The core generates this interrupt when a NAK is transmitted. In case of isochronous IN endpoints the interrupt gets generated when a zero length packet is transmitted due to unavailability of data in the TXFifo.	1'b0	R_SS_WC
12	BbleErr (Babble Error) interrupt (BbleErrIntrpt) The core generates this interrupt when babble is received for the endpoint.	1'b0	R_SS_WC
11	PktDrpSts (Packet Dropped Status) This bit indicates to the application that an ISOC OUT packet has been dropped. This bit does not have an associated mask bit and does not generate an interrupt. **Dependency**: This bit is valid in non Scatter/Gather DMA mode when periodic transfer interrupt feature is selected.	1'b0	R_SS_WC
10	Reserved		RO
9	BNA (Buffer Not Available) Interrupt (BNAIntr) The core generates this interrupt when the descriptor accessed is not ready for the Core to process, such as Host busy or DMA done. **Dependency**: This bit is valid only when Scatter/Gather DMA mode is enabled.	1'b0	R_SS_WC

8	FIFO Underrun (TxfifoUndrn) Applies to IN endpoints only The core generates this interrupt when it detects a transmit FIFO underrun condition for this endpoint. **Dependency**: This interrupt is valid only when both of the following conditions are true: ▪ Parameter OTG_EN_DED_TX_FIFO=1 ▪ Thresholding is enabled OUT Packet Error (OutPktErr) Applies to OUT endpoints only This interrupt is asserted when the core detects an overflow or a CRC error for an OUT packet. **Dependency**: This interrupt is valid only when both of the following conditions are true: ▪ Parameter OTG_EN_DED_TX_FIFO=1 ▪ Thresholding is enabled.	1'b0 R_SS_WC
7	Transmit FIFO Empty (TxFEmp) This bit is valid only for IN Endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core AHB Configuration register (GAHBCFG.NPTxFEmpLvl)).	1'b1	RO
6	N Endpoint NAK Effective (INEPNakEff) This bit should be cleared by writing a 1'b1 before writing a 1'b1 to corresponding DIEPCTLn.CNAK. The interrupt indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This interrupt does not guarantee that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit. This bit is applicable only when the endpoint is enabled. Back-to-Back SETUP Packets Received (Back2BackSETup) Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint. This bit is not valid in Slave mode.	1'b0 1'b0	R_SS_WC R_SS_WC
5	IN Token Received with EP Mismatch (INTknEPMis) Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. Status Phase Received For Control Write (StsPhseRcvd) This interrupt is	1'b0	R_SS_WC

	valid only for Control OUT endpoints and only in Scatter Gather DMA mode. This interrupt is generated only after the core has transferred all the data that the host has sent during the data phase of a control write transfer, to the system memory buffer. The interrupt indicates to the application that the host has switched from data phase to the status phase of a Control Write transfer. The application can use this interrupt to ACK or STALL the Status phase, after it has decoded the data phase. This is applicable only in case of Scatter Gather DMA mode.		
4	IN Token Received When TxFIFO is Empty (INTknTXFEmp) Indicates that an IN token was received when the associated TxFIFO (periodic/non- periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. OUT Token Received When Endpoint Disabled (OUTTknEPdis) Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received.	1'b0	R_SS_WC
3	Timeout Condition (TimeOUT) ▪ In shared TX FIFO mode, applies to non-isochronous IN endpoints only. ▪ In dedicated FIFO mode, applies only to Control IN endpoints. ▪ In Scatter/Gather DMA mode, the TimeOUT interrupt is not asserted. Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint. SETUP Phase Done (SetUp) Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet.	1'b0	R_SS_WC
2	AHB Error (AHBErr) Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an AHB error during an AHB read/write. The application can read the corresponding endpoint DMA address register to get the error address.	1'b0	R_SS_WC
1	Endpoint Disabled Interrupt (EPDisbld) Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.	1'b0	R_SS_WC
0	Transfer Completed Interrupt (XferCompl)	1'b0	R_SS_WC

	Applies to IN and OUT endpoints. ▪ When Scatter/Gather DMA mode is enabled ▪ For IN endpoint this field indicates that the requested data from the descriptor is moved from external system memory to internal FIFO. ▪ For OUT endpoint this field indicates that the requested data from the internal FIFO is moved to external system memory. This interrupt is generated only when the corresponding endpoint descriptor is closed, and the IOC bit for the corresponding descriptor is set. ▪ When Scatter/Gather DMA mode is disabled, this field indicates that the programmed transfer is complete on the AHB as well as on the USB, for this endpoint.		

25.5.5.24 Device Endpoint 0 Transfer Size Register (DIEPTSIZ0/DOEPTSIZ0)

✧ Offset for IN endpoints: 910h

✧ Offset for OUT endpoints: B10h

The application must modify this register before enabling endpoint 0. Once endpoint 0 is enabled using Endpoint Enable bit of the Device Control Endpoint 0 Control registers (DIEPCTL0.EPEna/DOEPCTL0.EPEna), the core modifies this register. The application can only read thisregister once the core has cleared the Endpoint Enable bit.

Nonzero endpoints use the registers for endpoints 1-15.

When Scatter/Gather DMA mode is enabled, this register must not be programmed by the application. If the application reads this register when Scatter/Gather DMA mode is enabled, the core returns all zeros.

Table 25-68 Device IN Endpoint 0 Transfer Size Register: DIEPTSIZ0

Field	Description	Reset	Access
31:21	Reserved		RO
20:19	Packet Count (PktCnt) Indicates the total number of USB packets that constitute the Transfer Size amount of data for endpoint 0. This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO.	2'b0	R_W
18:7	Reserved		RO
6:0	Transfer Size (XferSize) Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet from the external memory is written to the TxFIFO.	7'h0	R_W

Table 25-69 Device OUT Endpoint 0 Transfer Size Register: DOEPTSIZ0

Field	Description	Reset	Acces
31	Reserved		RO
30:29	SETUP Packet Count (SUPCnt) This field specifies the number of back-to-back SETUP data packets the endpoint can receive. ▪ 2'b01: 1 packet ▪ 2'b10: 2 packets ▪ 2'b11: 3 packets	2'h0	R_W
28:20	Reserved		RO
19	Packet Count (PktCnt) This field is decremented to zero after a packet is written into the RxFIFO.	1'b0	R_W
18:7	Reserved		RO
6:0	Transfer Size (XferSize) Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.	7'h0	R_W

25.5.5.25 Device Endpoint-*n* Transfer Size Register (DIEPTSIZn/DOEPTSIZn)

✧ Endpoint_number: 1 ≤n ≤15
✧ Offset for IN endpoints: 910h + (Endpoint_number * 20h)
✧ Offset for OUT endpoints: B10h + (Endpoint_number * 20h)

The application must modify this register before enabling the endpoint. Once the endpoint is enabled using Endpoint Enable bit of the Device Endpoint-n Control registers (DIEPCTLn.EPEna/DOEPCTLn.EPEna), the core modifies this register. The application can only read this register once the core has cleared the Endpoint Enable bit.

This register is used only for endpoints other than Endpoint 0.

Table 25-70 Device Endpoint-n Transfer Size Register: DIEPTSIZn/DOEPTSIZn

Field	Description	Reset	Access
31	Reserved		RO
30:29	Multi Count (MC) Applies to IN endpoints only. For periodic IN endpoints, this field indicates the number of packets that must be transmitted per microframe on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints. ▪ 2'b01: 1 packet ▪ 2'b10: 2 packets	2'b0	R_W

	▪ 2'b11: 3 packets For non-periodic IN endpoints, this field is valid only in Internal DMA mode. It specifies the number of packets the core must fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the Device Endpoint-*n* Control register (DIEPCTLn.NextEp). Received Data PID (RxDPID) Applies to isochronous OUT endpoints only. This is the data PID received in the last packet for this endpoint. ▪ 2'b00: DATA0 ▪ 2'b01: DATA2 ▪ 2'b10: DATA1 ▪ 2'b11: MDATA SETUP Packet Count (SUPCnt) Applies to control OUT Endpoints only. This field specifies the number of back-to-back SETUP data packets the endpoint can receive. ▪ 2'b01: 1 packet ▪ 2'b10: 2 packets ▪ 2'b11: 3 packets		RO RO R_W
28:19	Packet Count (PktCnt) Indicates the total number of USB packets that constitute the Transfer Size amount of data for this endpoint (PktCnt = XferSize / MPS). ▪ IN Endpoints: This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO. ▪ OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the RxFIFO.	10'h0	R_W
18:0	Transfer Size (XferSize) This field contains the transfer size in bytes for the current endpoint. The transfer size (XferSize) = Sum of buffer sizes across all descriptors in the list for the endpoint. In Buffer DMA, the core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. ▪ IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO. ▪ OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.	19'h0	R_W

Note: Note for Descriptor DMA

The maximum transfer size supported is 219 bytes

The maximum packet count supported = XferSize / Max Packet Size
The Transfer Size can span across multiple descriptors

25.5.5.26 Device Endpoint-*n* DMA Address Register (DIEPDMAn/DOEPDMAn)

- ✧ Endpoint_number: 0 ≤n ≤15
- ✧ Offset for IN endpoints: 914h + (Endpoint_number * 20h)
- ✧ Offset for OUT endpoints: B14h + (Endpoint_number * 20h)

These registers are implemented in RAM instead of flop-based implementation.

Table 25-71 Device Endpoint-n DMA Address Register: DIEPDMAn/DOEPDMAn

Field	Description	Reset	Access
31:0	DMA Address (DMAAddr) Holds the start address of the external memory for storing or fetching endpoint data. **Note**: For control endpoints, this field stores control OUT data packets as well as SETUP transaction data packets. When more than three SETUP packets are received back-to-back, the SETUP data packet in the memory is overwritten. This register is incremented on every AHB transaction. The application can give only a DWORD-aligned address. ▪ When Scatter/Gather DMA mode is not enabled, the application programs the start address value in this field. ▪ When Scatter/Gather DMA mode is enabled, this field indicates the base pointer for the descriptor list.	"X" if not programmed as the register is in SPRAM	R_W

25.5.5.27 Device Endpoint-*n* DMA Buffer Address Register (DIEPDMABn/DOEPDMABn)

- ✧ Endpoint_number: 0 ≤ n ≤15
- ✧ Offset for IN endpoints: 91Ch + (Endpoint_number * 20h)
- ✧ Offset for OUT endpoints: B1Ch + (Endpoint_number * 20h)

These fields are present only in case of Scatter/Gather DMA. These registers are implemented in RAM instead of flop-based implementation.

Table 25-72 Device Endpoint-n DMA Buffer Address Register: DIEPDMABn/DOEPDMABn

Field	Description	Reset	Access
31:0	DMA Buffer Address (DMABufferAddr) Holds the current buffer address.This register is updated as and when the data transfer for the corresponding end point is in progress.	"X" if not programmed as the	R_O

	This register is present only in Scatter/Gather DMA mode. Otherwise this field is reserved.	register is in SPRAM	

25.5.5.28 Device IN Endpoint Transmit FIFO Status Register (DTXFSTSn)

✧ Endpoint_number: 0 ≤ n ≤15

✧ Offset for IN endpoints: 918h + (Endpoint_number * 20h)

This read-only register contains the free space information for the Device IN endpoint TxFIFO.

Table 25-73 Device IN Endpoint Transmit FIFO Status Register: DTXFSTSn

Field	Description	Reset	Access
31:16	Reserved.		RO
15:0	IN Endpoint TxFIFO Space Avail (INEPTxFSpcAvail) Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. ▪ 16'h0: Endpoint TxFIFO is full ▪ 16'h1: 1 word available ▪ 16'h2: 2 words available ▪ 16'hn: n words available (where 0 ≤n ≤32,768) ▪ 16'h8000: 32,768 words available ▪ Others: Reserved	Configurable	RO

25.6 Operation Flow

25.6.1 Core Initialization

If the cable is connected during power-up, the Current Mode of Operation bit in the Core Interrupt register (GINTSTS.CurMod) reflects the mode. The DWC_otg core enters Host mode when an “A” plug is connected, or Device mode when a “B” plug is connected.

This section explains the initialization of the DWC_otg core after power-on. The application must follow this initialization sequence irrespective of whether the DWC_otg core is going to be configured in Host or Device mode of operations. All core global registers are initialized according to the core’s configuration. The parameters referred to in this section are the DWC_otg configuration parameters defined in the Databook.

1. Read the User Hardware Configuration registers (GHWCFG1, 2, 3, and 4) to find the configuration parameters selected for DWC_otg core.
2. Program the following fields in the Global AHB Configuration (GAHBCFG) register.
   - DMA Mode bit (applicable only when the OTG_ARCHITECTURE parameter is set to Internal/External DMA)
   - AHB Burst Length field (applicable only when the OTG_ARCHITECTURE parameter is set to Internal/External DMA)

- Global Interrupt Mask bit = 1
- Non-periodic TxFIFO Empty Level (can be enabled only when the core is operating in Slave mode as a host or as a device in Shared FIFO operation.)
- Periodic TxFIFO Empty Level (can be enabled only when the core is operating in Slave mode)

3. Program the following field in the Global Interrupt Mask (GINTMSK) register:
   - GINTMSK.RxFLvlMsk = 1'b0
4. Program the following fields in GUSBCFG register.
   - HNP Capable bit (applicable only when the OTG_MODE parameter is set to 0)
   - SRP Capable bit (not applicable when the OTG_MODE parameter is set to Device Only)
   - ULPI DDR Selection bit (applicable only when the OTG_HSPHY_INTERFACE parameter is selected for ULPI)
   - External HS PHY or Internal FS Serial PHY Selection bit (not applicable when "None" is selected for the OTG_FSPHY_INTERFACE parameter)
   - ULPI or UTMI+ Selection bit (not applicable when "None" is selected for the OTG_HSPHY_INTERFACE parameter)
   - PHY Interface bit (applicable only when the OTG_HSPHY_INTERFACE parameter is selected for UTMI+)
   - HS/FS TimeOUT Time-Out Calibration field
   - USB Turnaround Time field
5. The software must unmask the following bits in the GINTMSK register.
   - OTG Interrupt Mask
   - Mode Mismatch Interrupt Mask
6. If the GUID register is selected for implementation in coreConsultant, the software has the option of programming this register.
7. The software can read the GINTSTS.CurMod bit to determine whether the DWC_otg core is operating in Host or Device mode. The software then follows either the "Host Initialization" or "Device Initialization" sequence.

25.6.2 Programming the Device Core

25.6.2.1 Device Initialization

The application must meet the following conditions to set up the device core to handle traffic:

- In Slave mode, GINTMSK.NPTxFEmpMsk, and GINTMSK.RxFLvlMsk must be unset.
- In DMA mode, the GINTMSK.NPTxFEmpMsk, and GINTMSK.RxFLvlMsk interrupts must be masked.

The application must perform the following steps to initialize the core at device on, power on, or after a mode change from Host to Device.

1. Program the following fields in DCFG register.
   - DescDMA bit (applicable only if OTG_EN_DESC_DMA parameter is set to high)
   - Device Speed
   - NonZero Length Status OUT Handshake
   - Periodic Frame Interval (If Periodic Endpoints are supported)
2. Program the Device threshold control register. This is required only if you are using DMA mode and you are planning to enable thresholding.
3. Clear the DCTL.SftDiscon bit. The core issues a connect after this bit is cleared.
4. Program the GINTMSK register to unmask the following interrupts.
   - USB Reset
   - Enumeration Done
   - Early Suspend
   - USB Suspend
   - SOF
5. Wait for the GINTSTS.USBReset interrupt, which indicates a reset has been detected on the USB and lasts for about 10 ms. On receiving this interrupt, the application must perform the steps listed in "Initialization on USB Reset" .
6. Wait for the GINTSTS.EnumerationDone interrupt. This interrupt indicates the end of reset on the USB. On receiving this interrupt, the application must read the DSTS register to determine the enumeration speed and perform the steps listed in "Initialization on Enumeration Completion".

At this point, the device is ready to accept SOF packets and perform control transfers on control endpoint 0.

25.6.2.2 Initialization on USB Reset

This section describes what the application must do when it detects a USB reset.

1. Set the NAK bit for all OUT endpoints:
   - DOEPCTLn.SNAK = 1 (for all OUT endpoints)
2. Unmask the following interrupt bits:
   - DAINTMSK.INEP0 = 1 (control 0 IN endpoint)
   - DAINTMSK.OUTEP0 = 1 (control 0 OUT endpoint)
   - DOEPMSK.SETUP = 1
   - DOEPMSK.XferCompl = 1

- DIEPMSK.XferCompl = 1
- DIEPMSK.TimeOut = 1

3. To transmit or receive data, the device must initialize more registers as specified in "Device Initialization" .
4. Set up the Data FIFO RAM for each of the FIFOs (only if Dynamic FIFO Sizing is enabled)
   - Program the GRXFSIZ Register, to be able to receive control OUT data and setup data. If thresholding is not enabled, at a minimum, this must be equal to 1 max packet size of control endpoint 0 + 2 DWORDs (for the status of the control OUT data packet) + 10 DWORDs (for setup packets). If thresholding is enabled, at a minimum, this must be equal to 2 * (DTHRCTL.RxThrLen/4 + 1) + 2 DWORDs (for the status of the control OUT data packet) + 10 DWORDs (for setup packets)
   - Program the GNPTXFSIZ Register in Shared FIFO operation or dedicated FIFO size register (depending on the FIFO number chosen) in Dedicated FIFO operation, to be able to transmit control IN data. At a minimum, this must be equal to 1 max packet size of control endpoint 0.If thresholding is enabled, this can be programmed to less than one max packet size.
5. Reset the Device Address field in Device Configuration Register (DCFG).
6. (This step is not required if you are using Scatter/Gather DMA mode.) Program the following fields in the endpoint-specific registers for control OUT endpoint 0 to receive a SETUP packet
   - DOEPTSIZ0.SetUP Count = 3 (to receive up to 3 back-to-back SETUP packets)
   - In DMA mode, DOEPDMA0 register with a memory address to store any SETUP packets received

At this point, all initialization required to receive SETUP packets is done, except for enabling control OUT endpoint 0 in DMA mode.

25.6.2.3 Initialization on Enumeration Completion

This section describes what the application must do when it detects an Enumeration Done interrupt.

1. On the Enumeration Done interrupt (GINTSTS.EnumDone, read the DSTS register to determine the enumeration speed.
2. Program the DIEPCTL0.MPS field to set the maximum packet size. This step configures control endpoint 0. The maximum packet size for a control endpoint depends on the enumeration speed.
3. In DMA mode, program the DOEPCTL0 register to enable control OUT endpoint 0, to receive a SETUP packet. In Scatter/Gather DMA mode, the descriptors must be set up in memory before enabling the endpoint.
   - DOEPCTL0.EPEna = 1

4. Unmask the SOF interrupt.

At this point, the device is ready to receive SOF packets and is configured to perform control transfers on control endpoint 0.

25.6.3 Programming the Host Core

25.6.3.1 Host Initialization

To initialize the core as host, the application must perform the following steps.

1. Program GINTMSK.PrtInt to unmask.
2. Program the HCFG register to select full-speed host or high-speed host.
3. Program the HPRT.PrtPwr bit to 1'b1. This drives VBUS on the USB.
4. Wait for the HPRT0.PrtConnDet interrupt. This indicates that a device is connected to the port.
5. Program the HPRT.PrtRst bit to 1'b1. This starts the reset process.
6. Wait at least 10 ms for the reset process to complete.

25.6.3.2 Initializing Channels when the DWC_otg Host Core is in Buffer DMA or Slave Mode

To communicate with devices, the application must initialize and enable at least one channel. The method of initializing channels varies according to the mode of the DWC_otg Host core.

The following procedure for initializing channels is identical whether the DWC_otg Host core is operating in Buffer DMA or Slave Mode.

To initialize and enable a channel when the DWC_otg Host core is in Buffer DMA or Slave mode, the application must perform the following steps:

1. Program the GINTMSK register to unmask the following:
   - Non-periodic Transmit FIFO Empty for OUT transactions (applicable for Slave mode that operates in pipelined transaction-level with the Packet Count field programmed with more than one).
   - Non-periodic Transmit FIFO Half-Empty for OUT transactions (applicable for Slave mode that operates in pipelined transaction-level with the Packet Count field programmed with more than one).
2. Program the HAINTMSK register to unmask the selected channels' interrupts.
3. Program the HCINTMSK register to unmask the transaction-related interrupts of interest given in the Host Channel Interrupt register.
4. Program the selected channel's HCTSIZn register with the total transfer size, in bytes, and the expected number of packets, including short packets. The application must program the PID field with the initial data PID (to be used on the first OUT transaction or to be expected from the first IN transaction).
5. Program the Transfer Size field so that the channel's transfer size is a multiple of

the maximum packet size.

```
if (mps[epnum] mod 4) == 0
    transfer size[epnum] = n * (mps[epnum])          //Dword Aligned
 else
   transfer size[epnum] = n * (mps[epnum] + 4 - (mps[epnum] mod 4)) //Non Dword
Aligned
```

6. Program the selected channels' HCSPLTn register(s) with the hub and port addresses (split transactions only).
7. Program the HCCHARn register of the selected channel with the device's endpoint characteristics, such as type, speed, direction, and so forth. (The channel can be enabled by setting the Channel Enable bit to 1'b1 only when the application is ready to transmit or receive any packet).

25.6.3.3 Host Connection

This section explains the host connection flow when OTG_MODE configuration parameter is set to 0,1,2,5, or 6. Following steps explain the host connection flow:

1. When the USB Cable is plugged to the Host port, the core triggers GINTSTS.ConIDStsChng interrupt.
2. When the Host application detects GINTSTS.ConIDStsChng interrupt, the application can perform
   - Turn on VBUS by setting HPRT.PrtPwr = 1'b1 or
   - Wait for SRP Signaling from Device (when OTG_MODE = 0, 1, 5) to turn on VBUS.
3. The PHY indicates VBUS power-on by asserting utmi_vbusvalid.
4. When the Host Core detects the device connection, it triggers the Host Port Interrupt (GINTSTS.PrtInt) to the application.
5. When GINTSTS.PrtInt is triggered, the application reads the HPRT register to check if the
6. HPRT.Port Connect Detected (PrtConnDet) bit is set or not.

26 MAC

26.1 Features

- 10/100 Mbps operation
- Supports RMII PHY interfaces
- Supports VLAN and CRC
- Full-duplex and half-duplex operation
- MDIO Master interface for PHY device configuration and management
- remote wake-up frame and magic packet frame processing

Section 8

BOOT

27 XBurst Boot ROM Specification

The X1000 contains an internal 16KB boot ROM. The CPU boots from the boot ROM after reset.

27.1 Boot Select

The boot sequence of the X1000 is controlled by boot_sel [2:0]. The configuration is shown as follow:

Table 27-1 Boot Configuration of X1000

boot_sel[2]	boot_sel[1]	boot_sel[0]	Boot configuration
1	X	X	EXTCLK is 26MHz
0	X	X	EXTCLK is 24MHz
X	1	1	Boot from SFC0
X	0	1	Boot from MSC0
X	1	0	Boot from USB 2.0 device

X: means "Don't Care"

27.2 Boot Procedure

After reset, the boot program on the internal boot ROM executes as follows:

1 Disable all interrupts and read boot_sel[2:0] to determine the boot method.

2 If it is boot from MMC/SD card at MSC0, its function pins MSC0_D0, MSC0_CLK, MSC0_CMD are initialized, the boot program loads the 12KB code from MMC/SD card to tcsm and jump to it. Only one data bus which is MSC0_D0 is used. The clock EXTCLK/128 is used initially. When reading data, the clock EXTCLK/4 is used.

3 If it is boot from USB, a block of code will be received through USB cable connected with host PC and be stored in tcsm. Then branch to this area in tcsm.

4 If it is boot from SPI nor/nand at SFC, its function pins SFC_CLK,SFC_CE,SFC_DR,SFC_DT, SFC_WP, SFC_HOLD are initialized, the boot program loads the 12KB code from MMC/SD card to tcsm and jump to it.

NOTE: The X1000's tcsm is 16KB, its address is from 0xf4000000 to 0xf4004000.

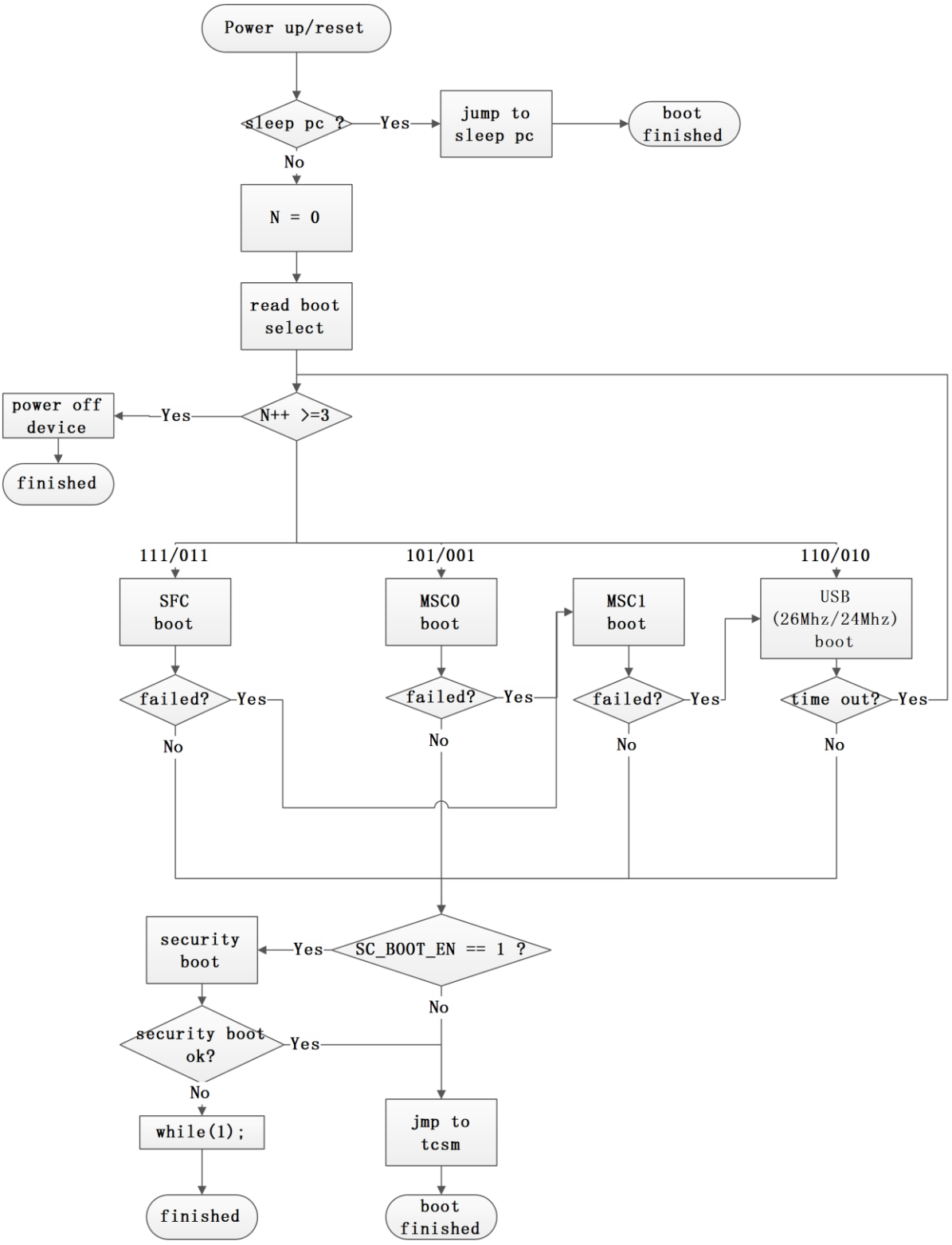


Figure 27-1 Boot sequence diagram of X1000

27.3 SPL Structure

For some special requirement, SPL(Second Program Loader) data needs to contain different information according to the different boot modes. The basic structure as shown below. SC_Boot KEY (1.5K) use for security boot, SPL signature is used to store some information for kinds of boot (512B), and SPL.text (not more than12K,and 512 byte aligned) is real code segment. the first 4k bytes of address space is used for stack and data space. all the boot model must jump to 0xf4001000 to execute SPL code and all the boot model contains the SPL structure show as Figure 27-2. Every boot model has its unique SPL signature,Introduces the SPL signature structure behind.

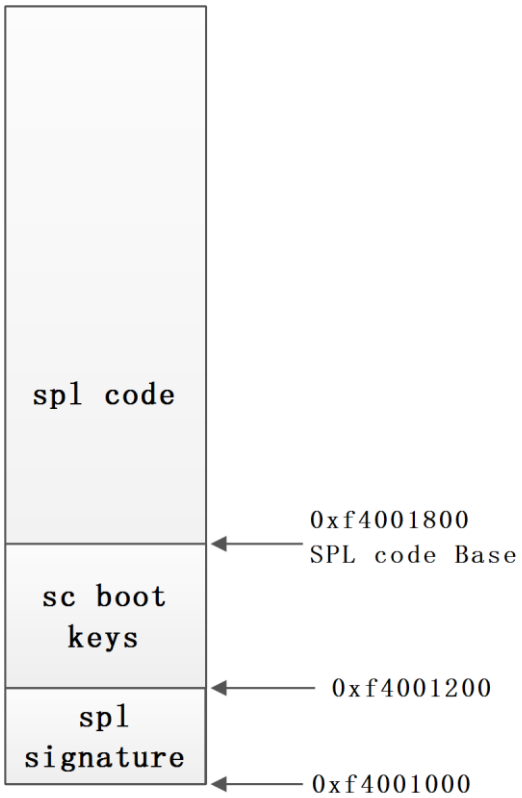


Figure 27-2 The basic SPL structure

27.4 SPL Paramaters

In order to support variable length of SPL size and increase the loading speed by changing the clock frequency. The SPL parameter is created to accomplish this function. This mechanism is only used for NAND/SD/MMC boot, and this mechanism is optional.

Parameters in a storage space in a certain structure is placed, and the size is 256 bytes .When sd card boot, the parameters will be placed at first 256 bytes space of the MBR. When NAND boot, the parameters will be placed at the end of the SPL signature. SPL Parameters's location and structure show below:

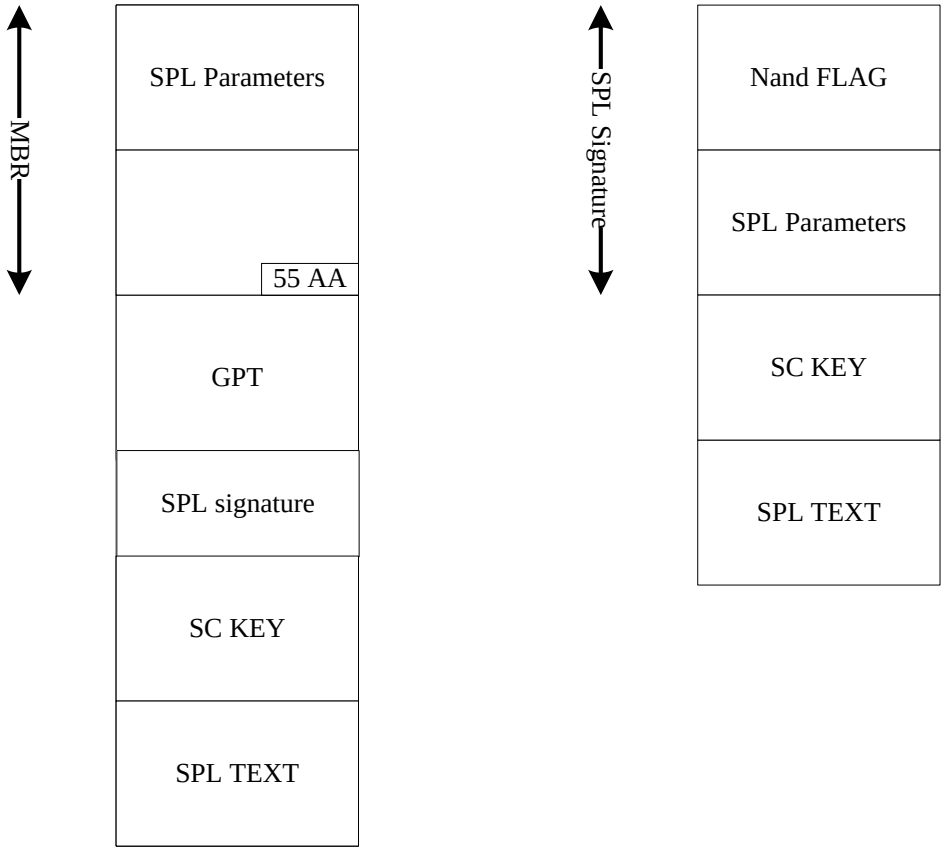


Figure 27-3 SPL parameters location

Table 27-2 SPL parameters structure

Offset	0x1 : 0x0	0x3 : 0x2	0x7 : 0x4	0xb : 0x8	0xf : 0xc
0x00	INGE		SPL length	CPU freq	Reserved
0x10	NAND Timing[0]		NAND Timing[1]	NAND Timing[2]	NAND Timing[3]
0x20	Desc[0].saddr	Desc[0].paddr	Desc[0].value	Desc[0].poll_h	Desc[0].poll_l
0x30	Desc[1].saddr	Desc[1].paddr	Desc[1].value	Desc[1].poll_h	Desc[1].poll_l
0x40	Desc[2].saddr	Desc[2].paddr	Desc[2].value	Desc[2].poll_h	Desc[2].poll_l
…	…	…	…	…	…
0xf0	Desc[13].saddr	Desc[13].paddr	Desc[13].value	Desc[13].poll_h	Desc[13].poll_l

Domain in SPL parameter structure:

INGE：

SPL Params ID，must store ‘I’ ‘N’ ‘G’ ‘E’, if this area is not store “INGE”,bootrom will use default params to boot.

SPL length：

The length of SPL in bytes, and must be 512 byte aligned. the size of SPL must be not more than 26K.

CPU freq：

Cpu frequency conversion value in HZ.

NAND Timing[0~3]：

After frequency conversion, the timing parameters of NAND.

```
struct {
        unsigned set_rw:8;
        unsigned wait_rw:8;
        unsigned hold_rw:8;
        unsigned set_cs:8;
        unsigned wait_cs:8;
        unsigned trr:8;
        unsigned tedo:8;
        unsigned trpre:8;
        unsigned twpre:8;
        unsigned tds:8;
        unsigned tdh:8;
        unsigned twpst:8;
        unsigned tdqsre:8;
        unsigned trhw:8;
        unsigned t1:8;
        unsigned t2:8;
    };
```

Descriptor：

A maximum of 14 groups descriptor, each descriptor is used to configure an cpm register.

The content of descriptor definition is as follows:

Saddr: Set Value Addr(This is lower 16 bits address , th higher 16bit address is 0xb000).

paddr: Polling Value Addr(This is lower 16 bits address , th higher 16bit address is 0xb000).

value: The configuration value of the “saddr” register

poll_h: Polling high bit mask(32bit).

poll_l: Polling low bit mask(32bit).

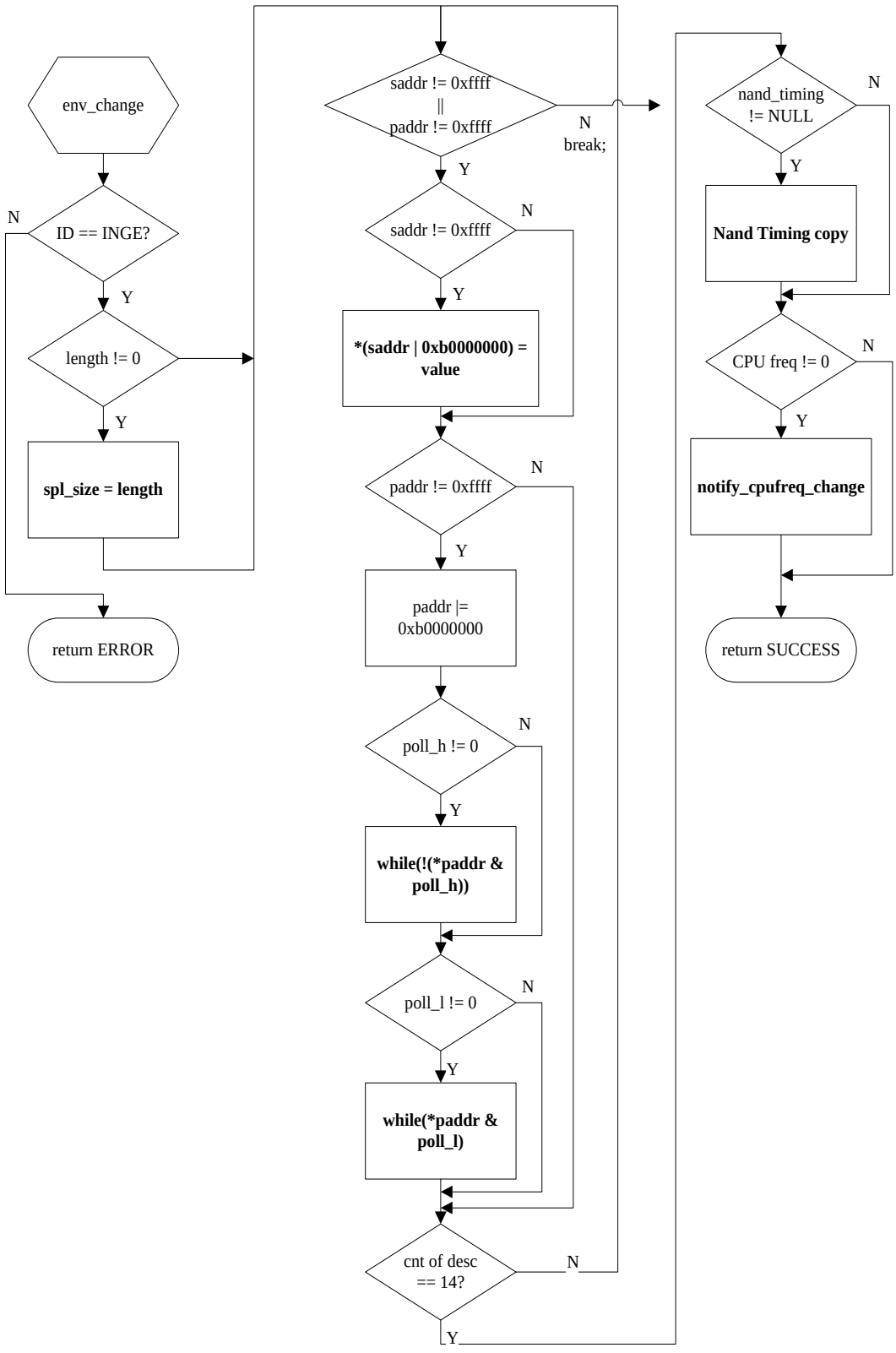


Figure 27-4 SPL configuration change Procedure

27.5 USB Boot Specification

When boot_sel[2:0] is selected as USB boot, the internal boot ROM downloads user program from the USB port to internal SRAM and branches to the internal SRAM to execute the program.

X1000 supports the external main crystal whose frequency is 24MHz.
The boot program supports both high-speed (480MHz) and full-speed (12MHz) transfer modes. The boot program uses the following two transfer types.

Table 27-3 Transfer Types Used by the Boot Program

Transfer Type	Description
Control Transfer	Used for transmitting standard requests and vendor requests.
Bulk Transfer	Used for responding to vendor requests and transmitting a user program.

Figure 27-5 shows an overview of the USB communication flow.

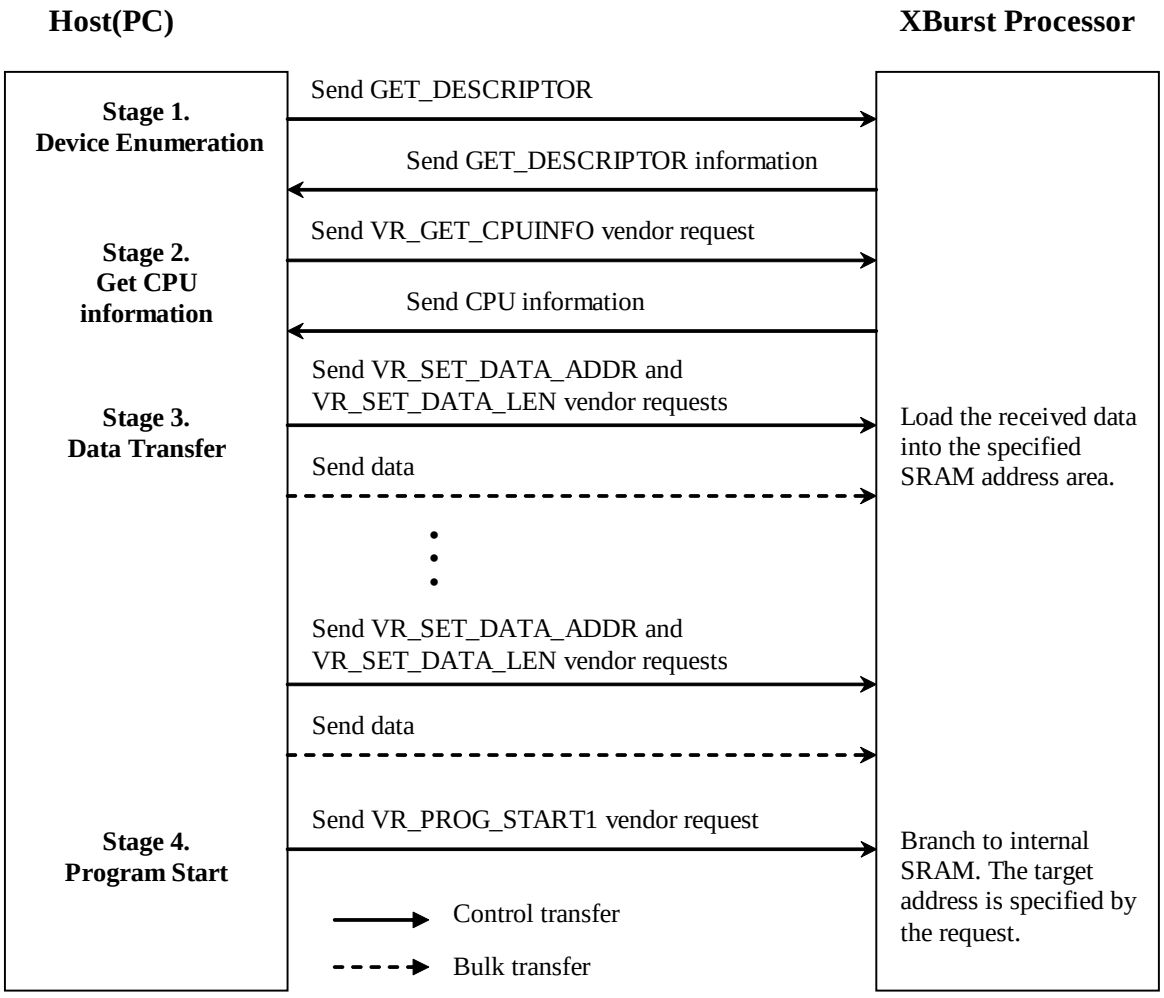


Figure 27-5 USB Communication Flow

The vendor ID and product ID for the USB boot device are 0xa108 and 0x1000 respectively. The Configuration for USB is for Control Endpoint 0 with Max Packet Size equals 64 bytes, Bulk IN at Endpoint 1 with Max Packet Size equals 512 bytes in high-speed and 64 bytes in full-speed, Bulk OUT at Endpoint 1 with Max Packet Size equals 512 bytes in high-speed and 64 bytes in full-speed.

The USB boot program provides six vendor requests through control endpoint for user to download/upload data to/from device, and to branch to a target address to execute user program. The six vendor requests are VR_GET_CPU_INFO (0x00), VR_SET_DATA_ADDRESS (0x01), VR_SET_DATA_LENGTH (0x02), VR_FLUSH_CACHES (0x03), VR_PROGRAM_START1 (0x04) and VR_PROGRAM_START2 (0x05). User program is transferred through Bulk IN or Bulk OUT endpoint.

When X1000 is reset with boot_sel[2:0] equals 111b or 001b ,the internal boot ROM will switch to USB boot mode and wait for USB requests from host. After connecting the USB device port to host, host will recognize the connection of a USB device, and start device enumeration. After finishing the device enumeration, user can send VR_GET_CPU_INFO (0x00) to query the device CPU information. If user wants to download/upload a program to/from device, two vendor requests VR_SET_DATA_ADDRESS (0x01) and VR_SET_DATA_LENGTH (0x02) should be sent first to tell the device the address and length in byte of the subsequent transferring data. Then data can be transferred through bulk-out/bulk-in endpoint. After this first stage program has been transferred to device, user can send vendor request VR_PROGRAM_START1 (0x04) to let the CPU to execute the program. This first stage program must not greater than 14KB and is normally used to init GPIO and SDRAM of the target board. At the end of the first stage program, it can return back to the internal boot ROM by jumping to ra ($31) register. Thus user can download a new program to the SDRAM of the target board like the first stage, and send vendor request VR_FLUSH_CACHES (0x03) and VR_PROGRAM_START2 (0x05) to let the CPU to execute the new program. Next figure is the typical procedure of USB boot.

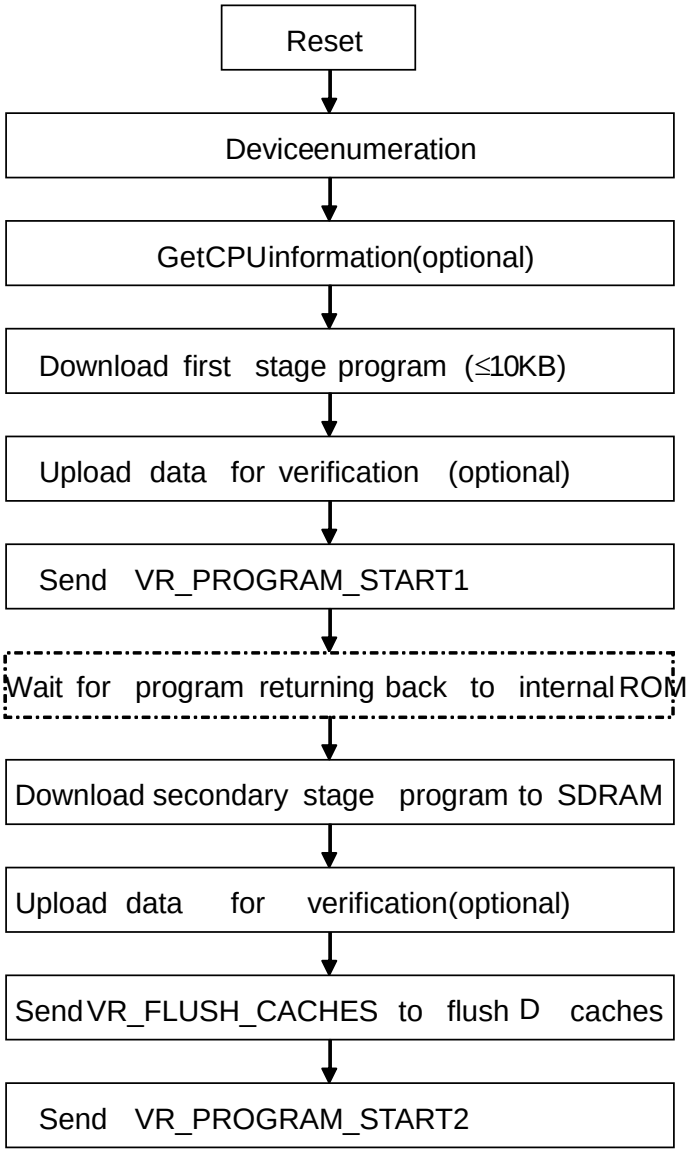


Figure 27-6 Typical Procedure of USB Boot

Following tables list all the vendor requests that USB boot program supports:

Table 27-4 Vendor Request 0 Setup Command Data Structure

Offset	Field	Size	Value	Description
0	bmRequestType	1	40H	D7 0: Host to Device. D6-D5 2: Vendor. D4-D0 0: Device.
1	bRequest	1	00H	VR_GET_CPU_INFO: get CPU information.
2	wValue	2	0000H	Not in used.
4	wIndex	2	0000H	Not in used.
6	wLength	2	0008H	8 bytes.

Table 27-5 Vendor Request 1 Setup Command Data Structure

Offset	Field	Size	Value	Description
0	bmRequestType	1	40H	D7 0: Host to Device. D6-D5 2: Vendor. D4-D0 0: Device.
1	bRequest	1	01H	VR_SET_DATA_ADDRESS: set address for next bulk-in/bulk-out transfer.
2	wValue	2	xxxxH	MSB (bit[31:16]) of the data address.
4	wIndex	2	xxxxH	LSB (bit[15:0]) of the data address.
6	wLength	2	0000H	Not in used.

Table 27-6 Vendor Request 2 Setup Command Data Structure

Offset	Field	Size	Value	Description
0	bmRequestType	1	40H	D7 0: Host to Device. D6-D5 2: Vendor. D4-D0 0: Device.
1	bRequest	1	02H	VR_SET_DATA_LENGTH: set length in byte for next bulk-in/bulk-out transfer.
2	wValue	2	xxxxH	MSB (bit[31:16]) of the data length.
4	wIndex	2	xxxxH	LSB (bit[15:0]) of the data length.
6	wLength	2	0000H	Not in used.

Table 27-7 Vendor Request 3 Setup Command Data Structure

Offset	Field	Size	Value	Description
0	bmRequestType	1	40H	D7 0: Host to Device. D6-D5 2: Vendor. D4-D0 0: Device.
1	bRequest	1	03H	VR_FLUSH_CACHES: flush I-Cache and D-Cache.

2	wValue	2	0000H	Not in used.
4	wIndex	2	0000H	Not in used.
6	wLength	2	0000H	Not in used.

Table 27-8 Vendor Request 4 Setup Command Data Structure

Offset	Field	Size	Value	Description
0	bmRequestType	1	40H	D7 0: Host to Device. D6-D5 2: Vendor. D4-D0 0: Device.
1	bRequest	1	04H	VR_PROGRAM_START1: transfer data from D-Cache to I-Cache and branch to address in I-Cache. **NOTE:** After downloading program from host to device for the first time, you can only use this request to start the program. Since the USB boot program will download data to D-Cache after reset. This request will transfer data from D-Cache to I-Cache and execute the program in I-Cache.
2	wValue	2	xxxxH	MSB (bit[31:16]) of the program entry point.
4	wIndex	2	xxxxH	LSB (bit[15:0]) of the program entry point.
6	wLength	2	0000H	Not in used.

Table 27-9 Vendor Request 5 Setup Command Data Structure

Offset	Field	Size	Value	Description
0	bmRequestType	1	40H	D7 0: Host to Device. D6-D5 2: Vendor. D4-D0 0: Device.
1	bRequest	1	05H	VR_PROGRAM_START2: branch to target address directly.
2	wValue	2	xxxxH	MSB (bit[31:16]) of the program entry point.
4	WIndex	2	xxxxH	LSB (bit[15:0]) of the program entry point.
6	WLength	2	0000H	Not in used.

27.6 MSC0 Boot Specification

All cards can boot from MMC/SD Boot from MSC0, the boot program will load 12KB code starting at address 0x4400 from MMC/SD card to cache. First the boot program initializes MSC0_D0, MSC0_CLK, MSC0_CMD as function pins. Only one data pin MSC0_D0 is used. Then the boot program sends CMD55 to test if it's SD or MMC card and initializes the card. At last it loads 12KB code

from the card to cpu tcsm and branches to execute the code in tcsm.
When initializing the card, the clock of EXTCLK/128 is used. And when reading data, the clock of EXTCLK/4 is used.
The procedure of the X1000 MMC/SD boot is shown as follow:

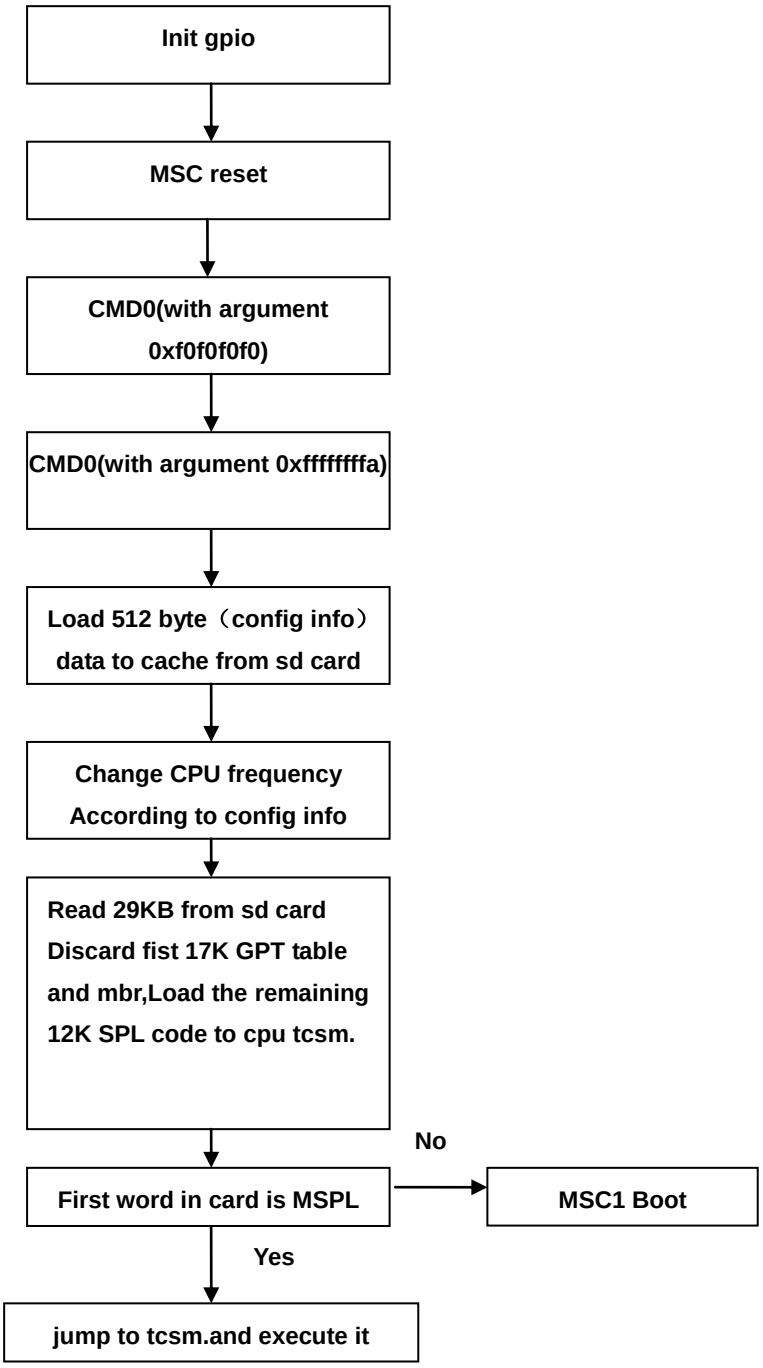


Figure 27-7 Typical Procedure of MSC0 Boot

SPL Structure show below , SPL signature fist 4 bytes is sd card magic code is used for Identification of SD,the remaining of the signature is unused and filled with 0xff. The first 512 byte of MBR store cpu

configuration information.

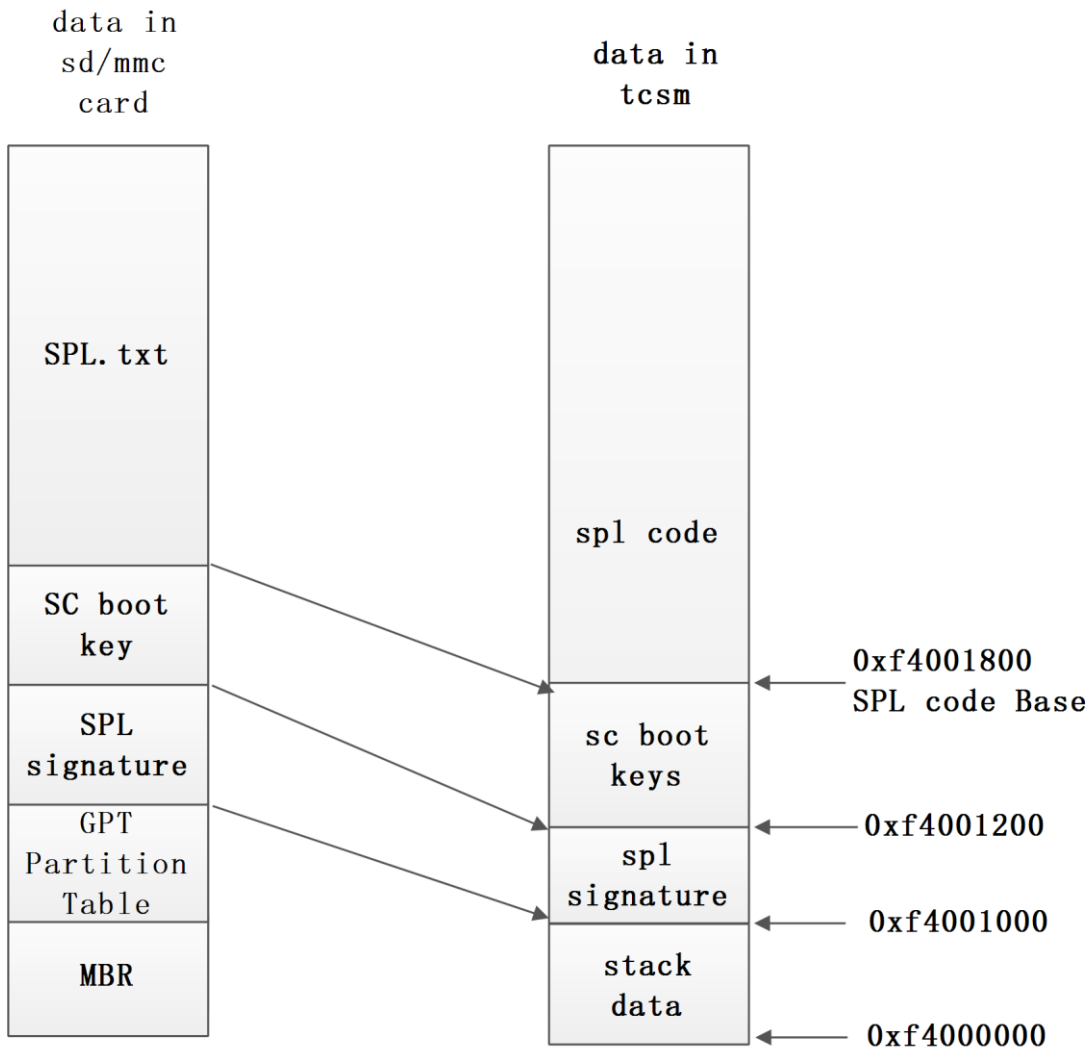


Figure 27-8 SPL Structure of MSC0/MSC1 Boot

27.7 MSC1 boot Specification

There is no msc1 boot support directly. But when boot from msc0 failed. Will try boot at msc1. Which is about the same procedure as msc0 boot. If msc1 boot failed, will goto usb boot.

27.8 SFC boot Specification

When boot_sel[2:0] is selected as SFC boot, CPU will goes into SFC boot，AT first cpu fetch a 16 bytes flag information table（The first 16 bytes in SPL signature） from the SPI device。CPU check the flag table to decide the SPI address length ， SPI device type（NOR flash or NAND flash），receive code length and the crc7 check data。SFC_boot detailed process shown “SFC Boot Procedure”

Note: Any irregularity in boot steps, SPI_boot will disable SFC controller and jump to MSC1 boot.

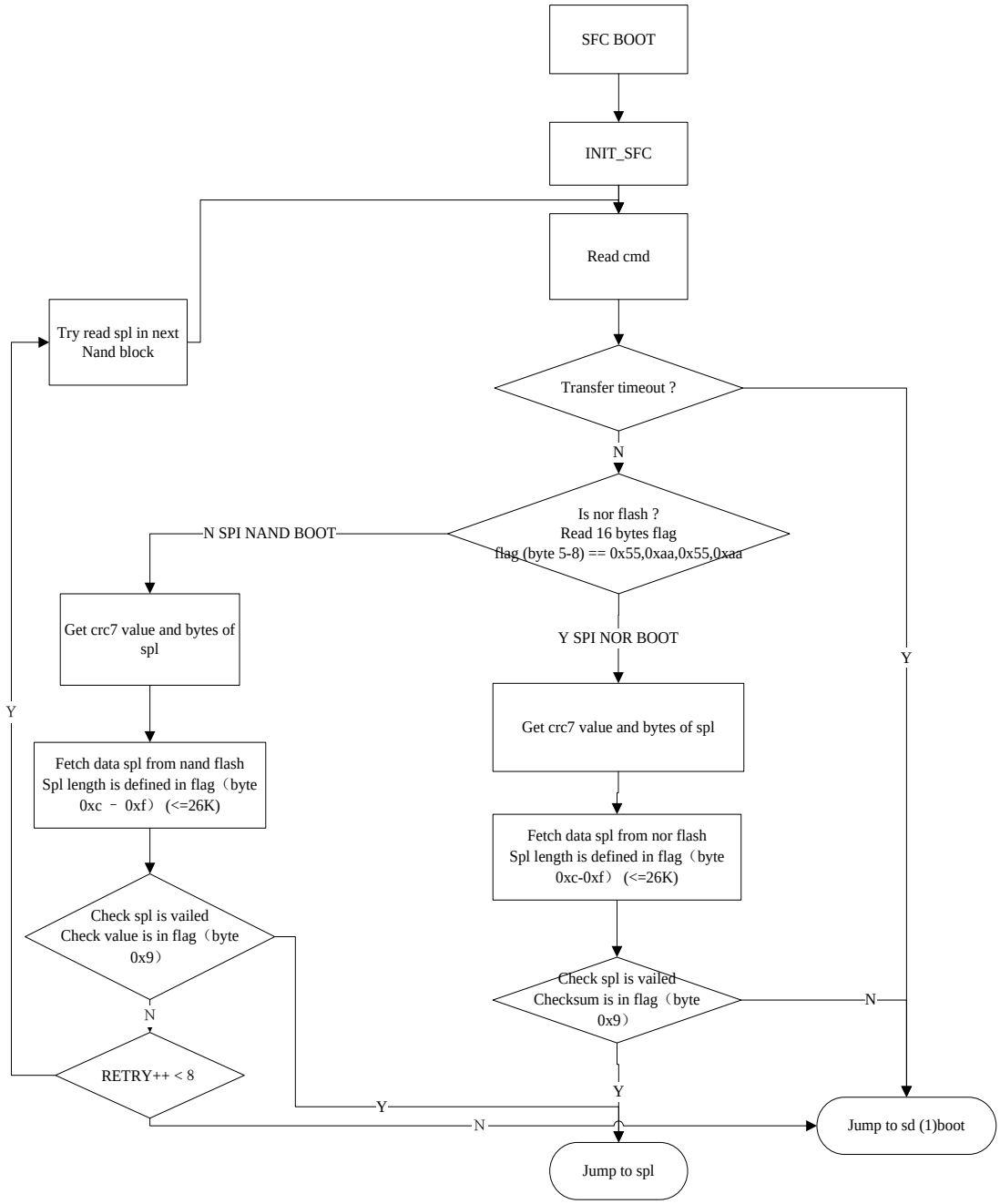


Figure 27-9 X1000 SPI Boot Procedure

SPI-NOR BOOT:

When SPI NOR flash by General-Purpose I/O Port-A 1,2,3,4 pin. In SPI NOR flash address 0x0~0xF, this space will store 16 bytes that agreed SPI boot flag.

The SPI boot flag information table and procedure of the T10 SPI NOR boot is shown as follow:

Table 27-10 SPI NOR flash boot flag informations(in spl signature)

Addr	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
Detail	0x06	0x05	0x04	0x03	0x02	0x55	0xaa	0x55
Addr	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
Detail	0xaa	Crc7	0x00	0x00	Len(LSB)	Len	Len	Len(MSB)

Explain:

0x00~0x04 :Must store 0x06,0x05,0x04,0x03,0x02

These part is used for cpu to decide the spi address length

0X05~0X08 :Must not be 0x55,0xAA,0x55,0xAA.

Magic code，device use this code decided whether is spi-nor flash boot

0x09 :The copy code's crc7 value .

0x0a ~0x0b :0x00

0x0C~0x0F :The length of copy data.

SPI-NAND BOOT:

When SPI NAND flash by General-Purpose I/O Port-A 1,2,3,4 pin. Like spi nor flash, in spi NAND flash address 0x0~0xF, this space will store 16 bytes that agreed SPI_boot flag.

The SPI_boot flag information table and procedure of the X1000 SPI NAND boot is shown as follow:

Table 27-11 SPI NAND flash boot flag informations(in spl signature)

Addr	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7
Detail	0x06	0x05	0x04	0x03	0x02	0x55	0xaa	0x55
Addr	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
Detail	0x00	Crc7	ppb	bpp	Len(LSB)	Len	Len	Len(MSB)

Explain:

0x00~0x04 :Must store 0x06,0x05,0x04,0x03,0x02

These part is used for cpu to decide the spi address length

0X05~0X08 :Must not be 0x55,0xAA,0x55,0x00.

Magic code，device use this code decided whether is spi-nand flash boot

0x09 :The copy code's crc7 value .

0x0a :The pages per block of SPI Nand, the value is X/32, for example, the pages per block is 64, then ppb is 64/32=2.

0x0b :The bytes per page of SPI Nand, the value is X/1024, for example, the bytes per page is 2048, then bpp is 2048/1024=2.

0x0C~0x0F :The length of copy data.