GR5526 Datasheet

Version: 1.2

Release Date: 2024-09-27

Shenzhen Goodix Technology Co., Ltd.

Contents

1 GR5526 Overview..... 1
1.1 Features..... 1
1.2 Block Diagram..... 4
1.3 Applications..... 6
2 Pinout..... 7
2.1 BGA83..... 7
2.2 QFN68..... 10
2.3 Pin Properties..... 14
3 MCU Core..... 17
3.1 Functional Overview..... 17
3.2 Interrupt..... 18
3.3 Memory Map..... 21
3.4 System Buses..... 24
3.5 Debug Interfaces..... 25
3.6 Memory Subsystem..... 25
3.6.1 SRAM..... 26
3.6.1.1 Introduction..... 26
3.6.2 eFuse..... 27
3.6.2.1 Introduction..... 27
3.6.2.2 Main Features..... 27
3.6.2.3 eFuse Information..... 28
3.6.2.4 Registers..... 30
3.6.2.4.1 EFUSE_TPGM..... 30
3.6.2.4.2 EFUSE_PGENB..... 31
3.6.2.4.3 EFUSE_OPTION..... 31
3.6.2.4.4 EFUSE_STAT..... 31
3.6.2.4.5 RF_REG_2..... 32
3.6.3 Flash (Non-Volatile Memory)..... 32
3.6.3.1 Write and Erase..... 32
3.6.3.2 Flash Operation Interfaces..... 33
3.6.3.3 Electrical Specifications..... 33
3.6.4 Cache..... 34
3.6.4.1 Registers..... 34
3.6.4.1.1 XF_CTRL..... 34
3.6.4.1.2 XF_HIT_COUNT..... 36
3.6.4.1.3 XF_MISS_COUNT..... 36
3.6.4.1.4 XF_STAT..... 36
3.6.5 PSRAM..... 37
3.6.5.1 Specifications..... 37

3.6.5.2 Features....38
3.6.5.3 Functional Description....38
3.6.5.4 Registers....40
3.6.5.4.1 MR0....40
3.6.5.4.2 MR1....42
3.6.5.4.3 MR2....42
3.6.5.4.4 MR3....43
3.6.5.4.5 MR4....43
3.6.5.4.6 MR6....44
3.6.5.4.7 MR8....45

4 PMU....46

4.1 Functional Overview....46
4.1.1 Power Management....48
4.1.1.1 Active Mode....49
4.1.1.2 IDLE Mode....50
4.1.1.3 Deep Sleep Mode....50
4.1.1.3.1 AON_SLP_CTRL....52
4.1.1.3.2 DPAD_LE_CTRL....52
4.1.1.4 Ultra Deep sleep mode....53
4.1.2 SRAM Power Control....53
4.1.2.1 Registers....54
4.1.2.1.1 MEM_PWR_SLP0....54
4.1.2.1.2 MEM_PWR_WKUP0....56
4.1.2.1.3 MEM_PWR_APPLY....59
4.1.2.1.4 MEM_PWR_STAT0....59
4.1.3 Wakeup Source....61
4.1.3.1 MCU_WAKEUP_CTRL....63
4.1.3.2 AON_SLP_EVENT....63
4.2 DC-DC Converter....64
4.2.1 RF_REG_4....65
4.3 DIG_CORE_LDO....65
4.3.1 PMU_DCORE_VREF....65
4.4 IO_LDO....66
4.4.1 RF_REG_0....66
4.5 BOD....66
4.5.1 Main Features....67
4.5.2 Functional Description....67
4.5.3 Registers....69
4.5.3.1 BOD_REG1....69
4.5.4 Specifications....70
4.6 Power Consumption....70
4.6.1 Electrical Specifications....71

4.6.1.1 Sleep....71
4.6.1.2 CPU Running....72
4.6.1.3 Radio Transmitting/Receiving....73
4.6.1.4 PSRAM....74
4.6.1.5 USB....75
4.6.1.6 ADC Active....75
4.6.1.7 TEMP Measure Active....75
4.6.1.8 VBATL Measure Active....75
4.6.1.9 COMP Active....75
4.6.1.10 BOD Active....76
4.6.1.11 TRNG Active....76
4.6.1.12 Security Active....76

5 Reset....77

5.1 Functional Overview....77
5.2 Power Reset....78
5.2.1 Power-on Reset....78
5.2.2 Brown-out Reset....78
5.3 Hardware Reset....78
5.3.1 External Reset Pin....79
5.3.2 Watchdog Reset....79
5.4 Software Reset....79
5.5 Reset Reason Recorder....79
5.5.1 Registers....79
5.5.2 Programming....80

6 Clocks....81

6.1 Functional Overview....81
6.2 High Frequency Clock....86
6.2.1 HFXO_32M....86
6.2.2 CPLL_192M....86
6.2.3 HFRC_192M....87
6.2.3.1 FS_REG_0....87
6.2.3.2 FS_REG_1....88
6.3 Low Frequency Clock....88
6.3.1 RNG_OSC....88
6.3.2 LFXO_32K....88
6.3.2.1 RF_REG_0....89
6.3.3 LFRC_32K....89
6.3.3.1 RC_RTC_REG_0....89
6.4 Clock Calibration....90
6.4.1 Operation....91
6.4.2 Registers....91

6.4.2.1 SL_CLK_CTRL....91
6.4.2.2 SL_CLK_CNT....92
6.4.2.3 SL_CLK_STAT....92
6.4.2.4 SL_CLK_CNT0....92
6.4.2.5 SL_CLK_CNT1....92
6.4.2.6 SL_CLK_INT_EN....93
6.4.2.7 SL_CLK_INT_CLR....93
6.4.2.8 SL_CLK_SEL....93
6.4.2.9 HS_CLK_CTRL....94
6.4.2.10 HS_CLK_CNT....94
6.4.2.11 HS_CLK_STAT....94
6.4.2.12 HS_CLK_CNT0....95
6.4.2.13 HS_CLK_CNT1....95
6.4.2.14 HS_CLK_INT_EN....95
6.4.2.15 HS_CLK_INT_CLR....95
6.4.2.16 HS_CLK_SEL....96
6.5 Registers....96
6.5.1 MCU_CLK_CTRL....96
6.5.2 AON_CLK....97
6.5.3 COMM_CTRL....98
6.5.4 ANA_SLP_CFG....98
6.5.5 XO_PLL_SET....99
6.5.6 XO_PLL_CLR....100
6.5.7 XO_PLL_STAT....100
6.5.8 MCU_SUBSYS_CG_CTRL_1....100
6.5.9 MCU_SUBSYS_CG_CTRL_2....102
6.5.10 MCU_PERIPH_CLK_OFF....103
6.5.11 MCU_PERIPH_CG_LP_EN....106
6.5.12 MCU_SUBSYS_SECU_CLK_CTRL....107
6.5.13 MCU_SUBSYS_MISC_CLK_CTRL....108
6.5.14 MCC_PLL_HFOSC_CLK_EN....109
6.5.15 MCC_CLK_FORCE_ON....109

7 Always-on Domain (AON)....111

7.1 Always-on Watchdog (AON WDT)....111
7.1.1 Introduction....111
7.1.2 Main Features....111
7.1.3 Functional Description....111
7.1.3.1 Block Diagram....112
7.1.3.2 Clock....112
7.1.3.3 Enable/Disable and Timer Reload....112
7.1.3.4 Alarm....112
7.1.4 Registers....112

7.1.4.1 CFG_0 113
7.1.4.2 LOCK 113
7.1.4.3 STAT 113
7.1.4.4 CLK 114
7.1.4.5 TIMER_WR 114
7.1.4.6 ALARM_WR 115
7.1.4.7 TIMER_RD 115
7.1.4.8 ALARM_RD 116
7.2 Real-Time Counter (RTC/Calendar) 116
7.2.1 Main Features 116
7.2.2 Functional Description 116
7.2.3 Registers 117
7.2.3.1 CTRL 117
7.2.3.2 CLK_DIV 118
7.2.3.3 IRQ_EN 118
7.2.3.4 IRQ_STAT 119
7.2.3.5 STAT 119
7.2.3.6 CLK_SEL 120
7.2.3.7 CMP_SET 120
7.2.3.8 TIMER_SET 120
7.2.3.9 TICK_SET 121
7.2.3.10 TIMER_RD 121
7.2.3.11 CMP_RD 121
7.2.3.12 TICK_RD 122
7.3 Sleep Timer (SLP TIMER) 122
7.3.1 Introduction 122
7.3.2 Main Features 122
7.3.3 Functional Description 122
7.3.4 Registers 123
7.3.4.1 CFG_0 123
7.3.4.2 STS 123
7.3.4.3 CLK 124
7.3.4.4 VAL_W 124
7.3.4.5 VAL_R 124

8 Peripherals 126

8.1 Pin Mux 126
8.1.1 Introduction 126
8.1.2 Functional Description 126
8.1.3 Registers 128
8.1.3.1 DPAD_MUX_CTRL_00_07 128
8.1.3.2 DPAD_MUX_CTRL_08_15 129
8.1.3.3 DPAD_MUX_CTRL_16_23 129

8.1.3.4 DPAD_MUX_CTRL_24_31 130
8.1.3.5 AON_PAD_MUX_CTRL 130
8.1.3.6 MSIO_A_PAD_MUX_CTRL 131
8.1.3.7 DPAD_MUX_CTRL_32_33 132
8.2 I/O 132
8.2.1 GPIO 132
8.2.1.1 Introduction 132
8.2.1.2 Main Features 133
8.2.1.3 Functional Description 133
8.2.1.4 Registers 134
8.2.1.4.1 DPAD_R_EN_BUS_0 134
8.2.1.4.2 DPAD_RTYP_BUS_0 134
8.2.1.4.3 DPAD_IN_EN_BUS_0 135
8.2.1.4.4 DPAD_R_EN_BUS_1 135
8.2.1.4.5 DPAD_RTYP_BUS_1 135
8.2.1.4.6 DPAD_IN_EN_BUS_1 136
8.2.1.4.7 DPAD_R_EN_MISC 136
8.2.1.4.8 DPAD_RTYP_MISC 137
8.2.1.4.9 DPAD_IN_EN_MISC 138
8.2.1.4.10 GPIOx_DATA 139
8.2.1.4.11 GPIOx_DATA_OUT 140
8.2.1.4.12 GPIOx_OUT_EN 140
8.2.1.4.13 GPIOx_OUT_CLR 141
8.2.1.4.14 GPIOx_INT_EN 141
8.2.1.4.15 GPIOx_INT_CLR 142
8.2.1.4.16 GPIOx_INT_TYPE_EN 142
8.2.1.4.17 GPIOx_INT_TYPE_CLR 143
8.2.1.4.18 GPIOx_INT_POL_EN 144
8.2.1.4.19 GPIOx_INT_POL_CLR 144
8.2.1.4.20 GPIOx_INT_STAT 145
8.2.1.5 Electrical Specifications 145
8.2.2 Always-On I/O 146
8.2.2.1 Introduction 146
8.2.2.2 Functional Description 146
8.2.2.3 Registers 146
8.2.2.3.1 AON_PAD_CTRL0 146
8.2.2.3.2 AON_PAD_CTRL1 147
8.2.2.3.3 AON_PAD_CLK 148
8.2.2.3.4 AON_PAD_MCU_OVR 148
8.2.2.4 Electrical Specifications 149
8.2.3 Mixed Signal I/O 149
8.2.3.1 Introduction 149

8.2.3.2 Functional Description..... 150
8.2.3.3 Registers..... 150
8.2.3.3.1 MSIO_A_PAD_CFG0..... 150
8.2.3.3.2 MSIO_A_PAD_CFG1..... 150
8.2.3.3.3 MSIO_A_PAD_MCU_OVR.....151
8.2.3.4 Electrical Specifications.....152
8.3 Timer Interfaces.....152
8.3.1 Timer..... 152
8.3.1.1 Introduction.....152
8.3.1.2 Main Features..... 152
8.3.1.3 Functional Description..... 153
8.3.1.4 Registers..... 153
8.3.1.4.1 CTRL..... 153
8.3.1.4.2 VAL..... 154
8.3.1.4.3 RELOAD..... 154
8.3.1.4.4 INT_STAT..... 154
8.3.2 Dual Timer..... 155
8.3.2.1 Introduction.....155
8.3.2.2 Main Features..... 155
8.3.2.3 Functional Description..... 155
8.3.2.4 Registers..... 156
8.3.2.4.1 RELOAD..... 156
8.3.2.4.2 VAL..... 156
8.3.2.4.3 CTRL..... 157
8.3.2.4.4 INT_CLR.....158
8.3.2.4.5 RAW_INT_STAT.....158
8.3.2.4.6 INT_STAT..... 158
8.3.2.4.7 BG_LOAD.....159
8.3.3 PWM..... 159
8.3.3.1 Introduction.....159
8.3.3.2 Main Features..... 159
8.3.3.3 Block Diagram..... 160
8.3.3.4 Functional Description..... 160
8.3.3.4.1 Operation.....162
8.3.3.4.2 Duty Cycle Calculation.....163
8.3.3.4.3 Recommended Setting in Breath Mode.....163
8.3.3.5 Registers..... 163
8.3.3.5.1 MODE.....163
8.3.3.5.2 UPDATE..... 164
8.3.3.5.3 PRD..... 166
8.3.3.5.4 CMPA0.....167
8.3.3.5.5 CMPA1.....167

8.3.3.5.6 CMPB0 ... 167
8.3.3.5.7 CMPB1 ... 167
8.3.3.5.8 CMPC0 ... 168
8.3.3.5.9 CMPC1 ... 168
8.3.3.5.10 AQ_CTRL ... 168
8.3.3.5.11 BR_PRD ... 170
8.3.3.5.12 HOLD ... 170
8.4 Communication Interfaces ... 171
8.4.1 UART ... 171
8.4.1.1 Introduction ... 171
8.4.1.2 Main Features ... 171
8.4.1.3 Block Diagram ... 171
8.4.1.4 Functional Description ... 173
8.4.1.4.1 UART (RS232) Serial Protocol ... 173
8.4.1.4.2 Fractional Baud Rate Support ... 174
8.4.1.4.3 FIFO Support ... 176
8.4.1.4.4 Clock Support ... 176
8.4.1.4.5 Interrupts ... 176
8.4.1.4.6 Auto Flow Control ... 178
8.4.1.4.7 Programmable THRE Interrupt ... 180
8.4.1.4.8 DMA Support ... 182
8.4.1.4.9 IrDA 1.0 SIR Protocol ... 182
8.4.1.5 Registers ... 184
8.4.1.5.1 RBR ... 184
8.4.1.5.2 DLL ... 184
8.4.1.5.3 THR ... 185
8.4.1.5.4 DLH ... 185
8.4.1.5.5 IER ... 186
8.4.1.5.6 FCR ... 187
8.4.1.5.7 IIR ... 189
8.4.1.5.8 LCR ... 190
8.4.1.5.9 MCR ... 192
8.4.1.5.10 LSR ... 194
8.4.1.5.11 MSR ... 195
8.4.1.5.12 SRBR ... 197
8.4.1.5.13 STHR ... 198
8.4.1.5.14 FAR ... 198
8.4.1.5.15 USR ... 199
8.4.1.5.16 TFL ... 200
8.4.1.5.17 RFL ... 201
8.4.1.5.18 SRR ... 201
8.4.1.5.19 SRTS ... 202

8.4.1.5.20 SBCR....202
8.4.1.5.21 SFE....203
8.4.1.5.22 SRT....204
8.4.1.5.23 STET....204
8.4.1.5.24 DLF....205
8.4.1.6 Electrical Specifications....205
8.4.2 SPI....206
8.4.2.1 Introduction....206
8.4.2.2 Main Features....206
8.4.2.3 Block Diagram....207
8.4.2.4 Functional Description....207
8.4.2.4.1 Motorola Serial Peripheral Interface....207
8.4.2.4.2 Transmit and Receive FIFO Buffers....209
8.4.2.4.3 SPI Interrupts....210
8.4.2.4.4 Transfer Modes....211
8.4.2.4.4.1 Transmit and Receive....211
8.4.2.4.4.2 Transmit Only....211
8.4.2.4.4.3 Receive Only....211
8.4.2.4.4.4 EEPROM Read....211
8.4.2.4.5 SPI Master Operation....212
8.4.2.4.6 SPI Slave Operation....212
8.4.2.5 Registers....213
8.4.2.5.1 CTRL0....213
8.4.2.5.2 CTRL1....217
8.4.2.5.3 SSI_EN....218
8.4.2.5.4 MW_CTRL....218
8.4.2.5.5 S_EN....220
8.4.2.5.6 BAUD....220
8.4.2.5.7 TX_FIFO_TL....221
8.4.2.5.8 RX_FIFO_TL....221
8.4.2.5.9 TX_FIFO_LV....222
8.4.2.5.10 RX_FIFO_LV....222
8.4.2.5.11 STAT....223
8.4.2.5.12 INT_MASK....225
8.4.2.5.13 INT_STAT....226
8.4.2.5.14 RAW_INT_STAT....227
8.4.2.5.15 TX_FIFO_OIC....228
8.4.2.5.16 RX_FIFO_OIC....229
8.4.2.5.17 RX_FIFO_UIC....229
8.4.2.5.18 MULTI_M_IC....230
8.4.2.5.19 INT_CLR....230
8.4.2.5.20 DMA_CTRL....231

8.4.2.5.21 DMA_TX_DL....231
8.4.2.5.22 DMA_RX_DL....232
8.4.2.5.23 DATA....232
8.4.2.5.24 RX_SMP_DLY....233
8.4.2.6 Electrical Specifications....233
8.4.2.6.1 SPIM Electrical Specifications....233
8.4.2.6.2 SPIS Electrical Specifications....234
8.4.3 I2C....235
8.4.3.1 Introduction....235
8.4.3.2 Main Features....235
8.4.3.3 Functional Description....236
8.4.3.3.1 I2C Operation....237
8.4.3.3.2 START and STOP Conditions....238
8.4.3.3.3 Addressing Slave....239
8.4.3.3.4 Transmitting and Receiving....240
8.4.3.3.5 SDA Hold Time....243
8.4.3.4 Registers....245
8.4.3.4.1 CTRL....245
8.4.3.4.2 TARGET_ADDR....248
8.4.3.4.3 S_ADDR....249
8.4.3.4.4 M_HS_ADDR....250
8.4.3.4.5 DATA_CMD....251
8.4.3.4.6 SS_CLK_HCNT....252
8.4.3.4.7 SS_CLK_LCNT....252
8.4.3.4.8 FS_CLK_HCNT....253
8.4.3.4.9 FS_CLK_LCNT....253
8.4.3.4.10 HS_CLK_HCNT....254
8.4.3.4.11 HS_CLK_LCNT....255
8.4.3.4.12 INT_STAT....255
8.4.3.4.13 INT_MASK....258
8.4.3.4.14 RAW_INT_STAT....260
8.4.3.4.15 RX_FIFO_THD....265
8.4.3.4.16 TX_FIFO_THD....265
8.4.3.4.17 CLR_INT....266
8.4.3.4.18 CLR_RX_UNDER....266
8.4.3.4.19 CLR_RX_OVER....267
8.4.3.4.20 CLR_TX_OVER....267
8.4.3.4.21 CLR_RD_REQ....267
8.4.3.4.22 CLR_TX_ABORT....268
8.4.3.4.23 CLR_RX_DONE....268
8.4.3.4.24 CLR_ACTIVITY....269
8.4.3.4.25 CLR_STOP_DET....269

8.4.3.4.26 CLR_START_DET.....270
8.4.3.4.27 CLR_GEN_CALL.....270
8.4.3.4.28 EN.....270
8.4.3.4.29 STAT.....272
8.4.3.4.30 TX_FIFO_LV.....274
8.4.3.4.31 RX_FIFO_LV.....274
8.4.3.4.32 SDA_HOLD.....275
8.4.3.4.33 TX_ABORT_SRC.....276
8.4.3.4.34 DMA_CTRL.....281
8.4.3.4.35 DMA_TX_LV.....282
8.4.3.4.36 DMA_RX_LV.....282
8.4.3.4.37 SDA_SETUP.....283
8.4.3.4.38 ACK_GEN_CALL.....283
8.4.3.4.39 EN_STAT.....284
8.4.3.4.40 FS_SPKLEN.....286
8.4.3.4.41 HS_SPKLEN.....287
8.4.3.4.42 SCL_STUCK_AT_LOW_TIMEOUT.....288
8.4.3.4.43 SDA_STUCK_AT_LOW_TIMEOUT.....288
8.4.3.4.44 CLR_SCL_STUCK_DET.....288
8.4.3.5 Electrical Specifications.....289
8.4.4 USB 2.0.....290
8.4.4.1 Introduction.....290
8.4.4.2 Main Features.....290
8.4.4.3 Functional Description.....291
8.4.4.3.1 Endpoint Configurations.....291
8.4.4.3.2 USB Enumeration Methods.....292
8.4.4.3.2.1 Automatic Hardware-enabled Enumeration.....292
8.4.4.3.2.2 Software-enabled Enumeration.....293
8.4.4.3.3 Bulk and Interrupt Transfer.....294
8.4.4.3.4 Isochronous Transfer.....296
8.4.4.4 Registers.....297
8.4.4.4.1 CTRL.....298
8.4.4.4.2 EP0_1_CTRL.....299
8.4.4.4.3 EP2_CTRL.....300
8.4.4.4.4 EP3_CTRL.....300
8.4.4.4.5 EP_ATTR.....301
8.4.4.4.6 INT_STAT.....302
8.4.4.4.7 INT_EN.....303
8.4.4.4.8 INT_CLR.....304
8.4.4.4.9 EP3_AHBM_RADDR.....306
8.4.4.4.10 EP3_AHMB_CTRL.....306
8.4.4.4.11 CTRL0.....306

8.4.4.4.12 EP3_XFER_LEN....308
8.4.4.4.13 RX_CNT....309
8.4.4.4.14 CFG_DESC_CTRL....309
8.4.4.4.15 STR_DESC0_CTRL....309
8.4.4.4.16 STR_DESC1_CTRL....310
8.4.4.4.17 EP0_FIFO_ADDR....310
8.4.4.4.18 EP1_FIFO_ADDR....310
8.4.4.4.19 EP2_FIFO_ADDR....311
8.4.4.4.20 EP3_FIFO_ADDR....311
8.4.4.4.21 SRAM_ADDR....311
8.4.4.4.22 STR_DESC2_CTRL....312
8.4.4.4.23 STR_DESC3_CTRL....312
8.4.4.4.24 EP4_CTRL....312
8.4.4.4.25 EP4_AHBM_RADDR....313
8.4.4.4.26 EP4_AHBM_CTRL....313
8.4.4.4.27 EP4_XFER_LEN....314
8.4.4.4.28 EP4_FIFO_WEN....314
8.4.4.4.29 EP4_FIFO_ADDR....314
8.4.4.4.30 EP5_CTRL....315
8.4.4.4.31 EP5_AHBM_RADDR....315
8.4.4.4.32 EP5_XFER_LEN....315
8.4.4.4.33 EP5_TIMER....316
8.4.4.4.34 EP5_FIFO_ADDR....316
8.4.4.4.35 EP5_RX_CNT....316
8.4.4.4.36 USB_DEBUG....317
8.4.4.4.37 USB_TRX_LDO....317
8.4.4.4.38 USB_LP_CTRL....318
8.4.4.4.39 USB_XCVR_CTRL....318
8.4.4.4.40 USB_SW_RST....319
8.4.4.5 Electrical Specifications....319
8.4.5 ISO/IEC 7816-3 Master....320
8.4.5.1 Introduction....320
8.4.5.2 Main Features....320
8.4.5.3 Functional Description....320
8.4.5.3.1 Block Diagram....320
8.4.5.3.2 Hardware Behavior....321
8.4.5.3.2.1 Physical Layer....321
8.4.5.3.2.1.1 CLK....321
8.4.5.3.2.1.2 RST....322
8.4.5.3.2.1.3 I/O....322
8.4.5.3.2.1.4 PRE....322
8.4.5.3.2.2 Power States....322

8.4.5.3.2.2.1 Activation on Cold Reset....322
8.4.5.3.2.2.2 Warm Reset....323
8.4.5.3.2.2.3 Clock Stop....323
8.4.5.3.2.2.4 Deactivation....323
8.4.5.3.2.3 Data Link Layer....324
8.4.5.3.2.3.1 Elementary Time Unit (ETU)....324
8.4.5.3.2.3.2 Character Frame....325
8.4.5.3.2.3.3 Error Signal and Character Repetition....325
8.4.5.3.2.3.4 Guard Time and Wait Time....325
8.4.5.3.2.4 Answer to Reset....326
8.4.5.3.2.5 Receive and Transmit....326
8.4.5.3.2.5.1 Byte DMA....326
8.4.5.3.2.5.2 Address Ranges....326
8.4.5.3.2.5.3 Receive Buffer....326
8.4.5.3.2.5.4 Transmit Buffer....327
8.4.5.3.2.5.5 DMA Error....327
8.4.5.3.2.6 Action State....327
8.4.5.3.2.7 Transport Layer....329
8.4.5.4 Registers....329
8.4.5.4.1 CTRL....329
8.4.5.4.2 STAT....331
8.4.5.4.3 CLK_CFG....334
8.4.5.4.4 TIMES_CFG....335
8.4.5.4.5 DATA_CFG....335
8.4.5.4.6 ADDR....336
8.4.5.4.7 START_ADDR....336
8.4.5.4.8 RX_END_ADDR....337
8.4.5.4.9 TX_END_ADDR....337
8.4.5.5 Electrical Specifications....338
8.5 Display Interfaces....338
8.5.1 DSPI....338
8.5.1.1 Introduction....338
8.5.1.2 Main Features....338
8.5.1.3 Functional Description....339
8.5.1.3.1 Protocols Introduction....339
8.5.1.3.1.1 3W1L Display Interface [A0+8-bit]....339
8.5.1.3.1.2 4W1L Display Interface [8-bit]....340
8.5.1.3.1.3 4W2L Display Interface [A0+8-bit]....340
8.5.1.3.2 Choice of DS and SPIMODE for Target Pixel Sizes....341
8.5.1.4 Registers....341
8.5.1.4.1 CTRL1....341
8.5.1.4.2 CTRL2....344

8.5.1.4.3 STAT....346
8.5.1.4.4 DATA....347
8.5.1.4.5 MODE....348
8.5.2 QSPI....349
8.5.2.1 Introduction....349
8.5.2.2 Main Features....349
8.5.2.3 Functional Description....349
8.5.2.3.1 Block Diagram....349
8.5.2.3.2 Clock Ratios....350
8.5.2.3.3 Motorola Serial Peripheral Interface....351
8.5.2.3.4 Transmit and Receive FIFO Buffers....351
8.5.2.3.5 QSPI Interrupts....351
8.5.2.3.6 Transfer Modes....352
8.5.2.3.6.1 Transmit and Receive....352
8.5.2.3.6.2 Transmit Only....352
8.5.2.3.6.3 Receive Only....352
8.5.2.3.6.4 EEPROM Read....353
8.5.2.3.7 DMA....353
8.5.2.3.8 Clock Stretching in Enhanced SPI Transfers....353
8.5.2.3.9 Memory Mapped Mode (XIP Mode)....354
8.5.2.3.9.1 Memory Mapped Read Model....354
8.5.2.3.9.2 Memory Mapped Write Model....354
8.5.2.3.9.3 Memory Mapped Space....355
8.5.2.3.9.4 Data Endian Mode for Memory Mapped Read Model....355
8.5.2.3.10 Enhanced Feature....356
8.5.2.3.10.1 Related Delay Between CS and SCK....356
8.5.2.4 Registers....356
8.5.2.4.1 CTRL0....356
8.5.2.4.2 CTRL1....360
8.5.2.4.3 QSPI_EN....361
8.5.2.4.4 MWC....361
8.5.2.4.5 SE....362
8.5.2.4.6 BAUD....363
8.5.2.4.7 TX_FTL....363
8.5.2.4.8 RX_FTL....364
8.5.2.4.9 TX_FL....365
8.5.2.4.10 RX_FL....365
8.5.2.4.11 STAT....366
8.5.2.4.12 INTMASK....367
8.5.2.4.13 INTSTAT....368
8.5.2.4.14 RAW_INTSTAT....370
8.5.2.4.15 TXEIC....371

8.5.2.4.16 RXOIC....371
8.5.2.4.17 RXUIC....372
8.5.2.4.18 MSTIC....372
8.5.2.4.19 INTCLR....373
8.5.2.4.20 DMAC....373
8.5.2.4.21 DMA_TDL....374
8.5.2.4.22 DMA_RDL....374
8.5.2.4.23 DATA....375
8.5.2.4.24 RX_SMP_DLY....375
8.5.2.4.25 SPI_CTRL0....375
8.5.2.4.26 XIP_MODE_BITS....377
8.5.2.4.27 XIP_INCR_INST....377
8.5.2.4.28 XIP_WRAP_INST....378
8.5.2.4.29 XIP_CTRL....378
8.5.2.4.30 XIP_SER....381
8.5.2.4.31 XIP_CNT_TIME_OUT....381
8.5.2.4.32 XIP_WR_INCR_INST....382
8.5.2.4.33 XIP_WR_WRAP_INST....382
8.5.2.4.34 XIP_WR_CTRL....383
8.5.2.4.35 QSPI_MODE_XIP....385
8.5.2.4.36 QSPI_CS_SETUP_DLY....386
8.5.2.4.37 QSPI_CS_RELEASE_DLY....387
8.5.2.5 Electrical Specifications....388
8.5.3 OSPI....389
8.5.3.1 Introduction....389
8.5.3.2 Main Features....389
8.5.3.3 Functional Description....390
8.5.3.3.1 Block Diagram....390
8.5.3.3.2 Control Interface....390
8.5.3.4 Registers....392
8.5.3.4.1 MEM_BASE_ADDR....392
8.5.3.4.2 MEM_TOP_ADDR....392
8.5.3.4.3 GLOBAL_RESET....392
8.5.3.4.4 ACCESS_TYPE....393
8.5.3.4.5 ACCESS_TIMING....393
8.5.3.4.6 DEEP_DOWN_CTRL....394
8.5.3.4.7 HALF_SLP_CTRL....395
8.5.3.4.8 INT_CTRL....396
8.5.3.4.9 XFER_STAT....397
8.5.3.4.10 CMD_CTRL_1....398
8.5.3.4.11 CMD_CTRL_2....398
8.5.3.4.12 DQS_TIMEOUT....399

8.5.3.4.13 PHY_CTRL_0....399
8.5.3.4.14 PHY_CTRL_1....400
8.5.3.4.15 PHY_CTRL_2....400
8.5.3.4.16 PHY_CTRL_3....400
8.5.3.4.17 PHY_CTRL_4....401
8.5.3.4.18 PHY_CTRL_5....401
8.5.3.4.19 PHY_CTRL_6....401
8.5.3.4.20 PHY_CTRL_7....402
8.5.3.4.21 RD_PREFETCH....402
8.5.4 DC....402
8.5.4.1 Introduction....402
8.5.4.2 Main Features....403
8.5.4.3 Block Diagram....403
8.5.4.4 Functional Description....404
8.5.4.4.1 Pin Mux....404
8.5.4.4.2 Clock Divider....404
8.5.4.4.3 Layer Overlays....405
8.5.4.4.4 Blending Modes....406
8.5.4.4.5 TSC Format....407
8.5.4.4.6 Display Interface....408
8.5.4.4.6.1 SPI 3-Wire/4-Wire Serial Interface....409
8.5.4.4.6.2 DualSPI Serial Interface....409
8.5.4.4.6.3 QuadSPI Serial Interface....409
8.5.4.4.7 Color Input Format....410
8.5.4.4.8 Color Output Format....413
8.5.4.4.8.1 SPI 3-Wire/4-Wire Output Format....414
8.5.4.4.8.1.1 Configuration 0x0C1 (RGB111: Option 0)....414
8.5.4.4.8.1.2 Configuration 0x0C9 (RGB111: Option 1)....415
8.5.4.4.8.1.3 Configuration 0x0D1 (RGB111: Option 2)....415
8.5.4.4.8.1.4 Configuration 0x0D9 (RGB111: Option 3)....415
8.5.4.4.8.1.5 Configuration 0x0E1 (RGB111: Option 4)....416
8.5.4.4.8.1.6 Configuration 0x0C2 (RGB332-Option 0)....416
8.5.4.4.8.1.7 Configuration 0x0C3 (RGB444-Option 0)....416
8.5.4.4.8.1.8 Configuration 0x0C5 (RGB565-Option 0)....417
8.5.4.4.8.1.9 Configuration 0x0C6 (RGB666-Option 0)....417
8.5.4.4.8.1.10 Configuration 0x0C7 (RGB888-Option 0)....417
8.5.4.4.8.2 DualSPI Output Format....418
8.5.4.4.8.2.1 Configuration 0x103 (RGB444-Option 0)....418
8.5.4.4.8.2.2 Configuration 0x10B (RGB444-Option 1)....418
8.5.4.4.8.2.3 Configuration 0x105 (RGB565-Option 0)....418
8.5.4.4.8.2.4 Configuration 0x106 (RGB666-Option 0)....418
8.5.4.4.8.2.5 Configuration 0x10E (RGB666-Option 1)....419

8.5.4.4.8.2.6 Configuration 0x107 (RGB888-Option 0)....419
8.5.4.4.8.2.7 Configuration 0x10F (RGB888-Option 1)....419
8.5.4.4.8.3 QuadSPI Serial Interface.... 419
8.5.4.4.8.3.1 Configuration 0x143 (RGB444-Option 0)....419
8.5.4.4.8.3.2 Configuration 0x145 (RGB565-Option 0)....420
8.5.4.4.8.3.3 Configuration 0x146 (RGB666-Option 0)....420
8.5.4.4.8.3.4 Configuration 0x147 (RGB888-Option 0)....420
8.5.4.5 Registers.... 420
8.5.4.5.1 MODE....420
8.5.4.5.2 CLK_DIV....421
8.5.4.5.3 BG_COLOR.... 422
8.5.4.5.4 RES.... 422
8.5.4.5.5 PLAY.... 423
8.5.4.5.6 INTERFACE_CFG.... 423
8.5.4.5.7 LAYER0_MODE.... 425
8.5.4.5.8 LAYER0_START....426
8.5.4.5.9 LAYER0_SIZE....427
8.5.4.5.10 LAYER0_ADDR.... 427
8.5.4.5.11 LAYER0_STRIDE.... 427
8.5.4.5.12 LAYER0_RES....428
8.5.4.5.13 LAYER1_MODE.... 428
8.5.4.5.14 LAYER1_START....430
8.5.4.5.15 LAYER1_SIZE....430
8.5.4.5.16 LAYER1_ADDR.... 431
8.5.4.5.17 LAYER1_STRIDE.... 431
8.5.4.5.18 LAYER1_RES....432
8.5.4.5.19 INTERFACE_CMD....432
8.5.4.5.20 INTERFACE_RDAT.... 433
8.5.4.5.21 CFG....433
8.5.4.5.22 ID....434
8.5.4.5.23 INT....435
8.5.4.5.24 STAT....435
8.5.4.5.25 COLOR_MODE....437
8.5.4.5.26 CRC....438
8.5.4.5.27 CLK_CTRL.... 438
8.5.4.5.28 FORMAT_CTRL.... 439
8.5.4.6 Electrical Specifications....440
8.5.5 GPU.... 441
8.5.5.1 Introduction....441
8.5.5.2 Main Features.... 441
8.5.5.3 Functional Description.... 443
8.5.5.3.1 Block Diagram....443

8.5.5.3.2 I/O Interfaces....443
8.5.5.3.3 Configuration Register File....444
8.5.5.3.4 Command List Processor....444
8.5.5.3.5 Rasterizer....444
8.5.5.3.6 Texture Map Unit....445
8.5.5.3.7 Fragment Processing Core....445
8.5.5.3.8 Render Output Unit....445
8.5.5.3.9 Framebuffer Compression....447
8.5.5.3.10 Color Modes....447
8.5.5.3.11 Color Expansion....453
8.5.5.3.12 Instruction Set Architecture....453
8.5.5.4 Registers....454
8.5.5.4.1 TEX0_BASE....454
8.5.5.4.2 TEX0_FSTRIDE....454
8.5.5.4.3 TEX0_RESXY....456
8.5.5.4.4 TEX1_BASE....456
8.5.5.4.5 TEX1_FSTRIDE....456
8.5.5.4.6 TEX1_RESXY....458
8.5.5.4.7 TEX_COLOR....458
8.5.5.4.8 TEX2_BASE....458
8.5.5.4.9 TEX2_FSTRIDE....459
8.5.5.4.10 TEX2_RESXY....460
8.5.5.4.11 TEX3_BASE....461
8.5.5.4.12 TEX3_FSTRIDE....461
8.5.5.4.13 TEX3_RESXY....462
8.5.5.4.14 BREAKPOINT....463
8.5.5.4.15 BREAKPOINT_MASK....463
8.5.5.4.16 CG_CTRL....463
8.5.5.4.17 DIRTYMIN....464
8.5.5.4.18 DIRTYMAX....464
8.5.5.4.19 IMEM_ADDR....465
8.5.5.4.20 IMEM_DATA_LOW....465
8.5.5.4.21 IMEM_DATA_HIGH....465
8.5.5.4.22 FLUSH_CTRL....466
8.5.5.4.23 CMD_STAT....466
8.5.5.4.24 RING_BUF_STORE....467
8.5.5.4.25 CMD_ADDR....467
8.5.5.4.26 CMD_SIZE....468
8.5.5.4.27 INT_CTRL....468
8.5.5.4.28 STAT....469
8.5.5.4.29 DRAW_CMD_NOHOLD....469
8.5.5.4.30 DRAW_STARTXY....470

8.5.5.4.31 DRAW_ENDXY....470
8.5.5.4.32 CLIPMIN....471
8.5.5.4.33 CLIPMAX....471
8.5.5.4.34 MAT_MCL....471
8.5.5.4.35 CODEPTR....472
8.5.5.4.36 DRAW_PT0_X....472
8.5.5.4.37 DRAW_PT0_Y....473
8.5.5.4.38 DRAW_COLOR....473
8.5.5.4.39 DRAW_PT1_X....473
8.5.5.4.40 DRAW_PT1_Y....474
8.5.5.4.41 BYPASS_ADDR....474
8.5.5.4.42 BYPASS_DATA....474
8.5.5.4.43 DRAW_PT2_X....475
8.5.5.4.44 DRAW_PT2_Y....475
8.5.5.4.45 CMD_LIST_ID....475
8.5.5.4.46 DRAW_PT3_X....476
8.5.5.4.47 DRAW_PT3_Y....476
8.5.5.4.48 CLIPMIN1....476
8.5.5.4.49 CLIPMAX1....477
8.5.5.4.50 MM00....477
8.5.5.4.51 MM01....477
8.5.5.4.52 MM02....478
8.5.5.4.53 MM10....478
8.5.5.4.54 MM11....478
8.5.5.4.55 MM12....479
8.5.5.4.56 MM20....479
8.5.5.4.57 MM21....479
8.5.5.4.58 MM22....480
8.5.5.4.59 DEPTH_START_L....480
8.5.5.4.60 DEPTH_START_H....480
8.5.5.4.61 DEPTH_DX_L....481
8.5.5.4.62 DEPTH_DX_H....481
8.5.5.4.63 DEPTH_DY_L....482
8.5.5.4.64 DEPTH_DY_H....482
8.5.5.4.65 RED_DX....482
8.5.5.4.66 RED_DY....483
8.5.5.4.67 GRE_DX....483
8.5.5.4.68 GRE_DY....483
8.5.5.4.69 BLU_DX....483
8.5.5.4.70 BLU_DY....484
8.5.5.4.71 ALF_DX....484
8.5.5.4.72 ALF_DY....484

8.5.5.4.73 RED_INIT....485
8.5.5.4.74 GRE_INIT....485
8.5.5.4.75 BLU_INIT....485
8.5.5.4.76 ALF_INIT....486
8.5.5.4.77 ROPBLENDER_BLEND_MODE....486
8.5.5.4.78 ROPBLENDER_DST_CKEY....487
8.5.5.4.79 ROPBLENDER_CONST_COLOR....487
8.5.5.4.80 IP_VERSION....488
8.5.5.4.81 IDREG....488
8.5.5.4.82 CFG....488
8.5.5.4.83 CFG_H....489
8.5.5.4.84 C0_REG....490
8.5.5.4.85 C1_REG....490
8.5.5.4.86 C2_REG....490
8.5.5.4.87 C3_REG....491
8.5.5.4.88 COORD_MASK....491
8.5.5.4.89 IRQ_ID....491
8.5.5.4.90 GP_FLAGS....492
8.5.5.4.91 SYS_INT....493
8.5.5.4.92 BUSERROR_MASK....494
8.6 Audio Interface....494
8.6.1 PDM....494
8.6.1.1 Introduction....494
8.6.1.2 Main Features....495
8.6.1.3 Block Diagram....495
8.6.1.4 Protocols Introduction....496
8.6.1.4.1 Protocols Introduction....496
8.6.1.4.2 PCM Introduction....497
8.6.1.5 Functional Description....498
8.6.1.5.1 PDM Pin Mux....498
8.6.1.5.2 PDM Operation....498
8.6.1.5.3 PDM Gain....498
8.6.1.5.4 PDM Enable....499
8.6.1.5.5 PDM Interrupt....500
8.6.1.5.6 PDM DMA....500
8.6.1.5.7 Hardware Connection Example....500
8.6.1.6 Registers....501
8.6.1.6.1 EN_LEFT....501
8.6.1.6.2 PGA_CFG_LEFT....502
8.6.1.6.3 DATA_LEFT....502
8.6.1.6.4 INT_LEFT....503
8.6.1.6.5 EN_RIGHT....503

8.6.1.6.6 PGA_CFG_RIGHT....504
8.6.1.6.7 DATA_RIGHT....504
8.6.1.6.8 INT_RIGHT....505
8.6.1.6.9 DATA....505
8.6.1.6.10 CLK....505
8.6.1.7 Electrical Specifications....506
8.6.2 I2S....507
8.6.2.1 Introduction....507
8.6.2.2 Main Features....507
8.6.2.3 Protocols Introduction....508
8.6.2.3.1 Signals and Formats....508
8.6.2.3.2 Sampling and CLK frequency....509
8.6.2.4 Functional Description....509
8.6.2.4.1 I2S Pin Mux....509
8.6.2.4.2 Master and Slave....509
8.6.2.4.3 Clock Divider....510
8.6.2.4.4 I2S Enable....511
8.6.2.4.5 I2S as Transmitter....512
8.6.2.4.6 I2S as Receiver....512
8.6.2.4.7 I2S FIFO....513
8.6.2.4.8 I2S DMA....514
8.6.2.5 Registers....514
8.6.2.5.1 EN....514
8.6.2.5.2 RX_EN....514
8.6.2.5.3 TX_EN....515
8.6.2.5.4 CLK_EN....515
8.6.2.5.5 SCLK_CFG....516
8.6.2.5.6 RX_FIFO_RST....517
8.6.2.5.7 TX_FIFO_RST....517
8.6.2.5.8 LEFT_RX_BUF....518
8.6.2.5.9 LEFT_TX_HDG....518
8.6.2.5.10 RIGHT_RX_BUF....519
8.6.2.5.11 RIGHT_TX_HDG....520
8.6.2.5.12 RX_CH_EN....520
8.6.2.5.13 TX_CH_EN....521
8.6.2.5.14 RX_CFG....521
8.6.2.5.15 TX_CFG....522
8.6.2.5.16 INT_STAT....523
8.6.2.5.17 INT_MASK....524
8.6.2.5.18 RX_OVER....525
8.6.2.5.19 TX_OVER....525
8.6.2.5.20 RX_FIFO_CFG....526

8.6.2.5.21 TX_FIFO_CFG....527
8.6.2.5.22 RX_FIFO_FLUSH....528
8.6.2.5.23 TX_FIFO_FLUSH....528
8.6.2.5.24 RX_DMA....529
8.6.2.5.25 RST_RX_DMA....529
8.6.2.5.26 TX_DMA....530
8.6.2.5.27 RST_TX_DMA....530
8.6.2.5.28 CLK_CFG....531
8.6.2.6 Electrical Specifications....531
8.7 Security Interfaces....532
8.7.1 KEYRAM....532
8.7.2 Advanced Encryption Standard (AES)....532
8.7.2.1 Introduction....532
8.7.2.2 Main Features....533
8.7.2.3 Functional Description....533
8.7.2.3.1 Standard....533
8.7.2.3.2 Module Overview....533
8.7.2.3.3 Usage....534
8.7.2.4 Registers....535
8.7.2.4.1 CTRL....535
8.7.2.4.2 CFG....536
8.7.2.4.3 STAT....537
8.7.2.4.4 INT....538
8.7.2.4.5 XFE_SIZE....539
8.7.2.4.6 RD_START_ADDR....539
8.7.2.4.7 WR_START_ADDR....539
8.7.2.4.8 KEY_ADDR....540
8.7.2.4.9 DATA_OUT0....540
8.7.2.4.10 DATA_OUT1....540
8.7.2.4.11 DATA_OUT2....541
8.7.2.4.12 DATA_OUT3....541
8.7.2.4.13 KEY0....541
8.7.2.4.14 KEY1....542
8.7.2.4.15 KEY2....542
8.7.2.4.16 KEY3....542
8.7.2.4.17 KEY4....543
8.7.2.4.18 KEY5....543
8.7.2.4.19 KEY6....543
8.7.2.4.20 KEY7....544
8.7.2.4.21 INIT_SSI....544
8.7.2.4.22 INIT_SSO....544
8.7.2.4.23 MASK_SSI....545

8.7.2.4.24 MASK_SSO 545
8.7.2.4.25 INIT_V0 545
8.7.2.4.26 INIT_V1 546
8.7.2.4.27 INIT_V2 546
8.7.2.4.28 INIT_V3 546
8.7.2.4.29 DATA_IN0 547
8.7.2.4.30 DATA_IN1 547
8.7.2.4.31 DATA_IN2 547
8.7.2.4.32 DATA_IN3 548
8.7.2.4.33 KEYPORT_MASK 548
8.7.3 Hash-based Message Authentication Code (HMAC) 548
8.7.3.1 Introduction 548
8.7.3.2 Main Features 549
8.7.3.3 Functional Description 549
8.7.3.3.1 Standard 549
8.7.3.3.2 Module Overview 549
8.7.3.3.3 Usage 550
8.7.3.4 Registers 551
8.7.3.4.1 CTRL 551
8.7.3.4.2 CFG 551
8.7.3.4.3 STAT 552
8.7.3.4.4 XFER_SIZE 553
8.7.3.4.5 INT 554
8.7.3.4.6 RD_START_ADDR 554
8.7.3.4.7 WR_START_ADDR 555
8.7.3.4.8 USER_HASH_0 555
8.7.3.4.9 USER_HASH_1 555
8.7.3.4.10 USER_HASH_2 556
8.7.3.4.11 USER_HASH_3 556
8.7.3.4.12 USER_HASH_4 556
8.7.3.4.13 USER_HASH_5 557
8.7.3.4.14 USER_HASH_6 557
8.7.3.4.15 USER_HASH_7 558
8.7.3.4.16 DATA_OUT 558
8.7.3.4.17 DATA_IN 558
8.7.3.4.18 KEY0 559
8.7.3.4.19 KEY1 559
8.7.3.4.20 KEY2 560
8.7.3.4.21 KEY3 560
8.7.3.4.22 KEY4 560
8.7.3.4.23 KEY5 561
8.7.3.4.24 KEY6 561

8.7.3.4.25 KEY7 561
8.7.3.4.26 KEY_ADDR 562
8.7.3.4.27 KEYPORT_MASK 562
8.7.4 Public Key Cryptography (PKC) 562
8.7.4.1 Introduction 562
8.7.4.2 Main Features 563
8.7.4.3 Functional Description 563
8.7.4.3.1 Standard 563
8.7.4.3.2 Module Overview 563
8.7.4.3.3 Usage 564
8.7.4.4 Registers 566
8.7.4.4.1 CTRL 566
8.7.4.4.2 CFG0 566
8.7.4.4.3 CFG1 567
8.7.4.4.4 CFG2 567
8.7.4.4.5 CFG3 568
8.7.4.4.6 CFG4 568
8.7.4.4.7 CFG5 568
8.7.4.4.8 CFG6 569
8.7.4.4.9 CFG7 569
8.7.4.4.10 CFG8 570
8.7.4.4.11 CFG9 570
8.7.4.4.12 CFG10 571
8.7.4.4.13 CFG11 571
8.7.4.4.14 CFG12 571
8.7.4.4.15 CFG13 572
8.7.4.4.16 SW_CTRL 572
8.7.4.4.17 SW_CFG0 573
8.7.4.4.18 SW_CFG1 573
8.7.4.4.19 SW_CFG2 574
8.7.4.4.20 SW_CFG3 574
8.7.4.4.21 SW_CFG4 575
8.7.4.4.22 SW_CFG5 575
8.7.4.4.23 SW_CFG6 576
8.7.4.4.24 SW_CFG7 576
8.7.4.4.25 SW_CFG8 576
8.7.4.4.26 SW_CFG9 577
8.7.4.4.27 SW_CFG10 577
8.7.4.4.28 SW_CFG11 577
8.7.4.4.29 SW_CFG12 578
8.7.4.4.30 SW_CFG13 578
8.7.4.4.31 INT_STAT 579

8.7.4.4.32 INT_EN....579
8.7.4.4.33 STAT....580
8.7.5 True Random Number Generator (TRNG)....580
8.7.5.1 Introduction....580
8.7.5.2 Main Features....580
8.7.5.3 Functional Description....580
8.7.5.3.1 Standard....580
8.7.5.3.2 Module Overview....580
8.7.5.3.3 Usage....582
8.7.5.4 Registers....583
8.7.5.4.1 CTRL....583
8.7.5.4.2 STAT....583
8.7.5.4.3 DATA....584
8.7.5.4.4 MCULOCK....584
8.7.5.4.5 LONG_RUN_STAT....584
8.7.5.4.6 CFG....585
8.7.5.4.7 SRC_CFG....586
8.7.5.4.8 FRO_CFG....587
8.7.5.4.9 USER_SEED....587
8.7.5.4.10 LONG_RUN_CFG....588
8.8 Analog Interfaces....588
8.8.1 13-Bit SADC....588
8.8.1.1 Introduction....588
8.8.1.2 Main Features....589
8.8.1.3 Block Diagram....589
8.8.1.4 Resource Sharing....590
8.8.1.5 Functional Description....590
8.8.1.5.1 SADC Raw Data Stream....591
8.8.1.5.2 SADC Sample Operation Mode....591
8.8.1.5.3 SADC Clock and Sampling Rate....592
8.8.1.5.4 Considerations for Using SADC....592
8.8.1.6 Measuring MSIO Input....593
8.8.1.7 Measuring Battery Voltage....593
8.8.1.8 Measuring Internal Temperature....594
8.8.1.9 SADC Manual Calibration Method....595
8.8.1.10 Registers....598
8.8.1.10.1 CFG....598
8.8.1.10.2 FIFO_RD....600
8.8.1.10.3 FIFO_CTRL....601
8.8.1.10.4 FIFO_STAT....601
8.8.1.10.5 CLK....602
8.8.1.10.6 ADC_GET_TKN_SW....602

8.8.1.10.7 ADC_RET_TKN_SW....603
8.8.1.10.8 ADC_GET_TKN_STAT....603
8.8.1.11 Electrical Specifications....604
8.8.2 Temperature Sensor....604
8.8.2.1 Introduction....604
8.8.2.2 Main Features....604
8.8.2.3 Functional Description....605
8.8.2.3.1 Acquisition Parameter Configuration....605
8.8.2.3.2 Conversion Formula....605
8.8.2.3.3 Measuring Internal Temperature....605
8.8.2.4 Registers....606
8.8.2.5 Electrical Specifications....606
8.8.3 Comparator....607
8.8.3.1 Introduction....607
8.8.3.2 Main Features....607
8.8.3.3 Functional Description....607
8.8.3.3.1 Overview....608
8.8.3.3.2 Hysteresis....609
8.8.3.3.3 Sample Clock....610
8.8.3.3.4 Res Degeneration....610
8.8.3.3.5 CascRes Half....611
8.8.3.3.6 Bias Current Set....611
8.8.3.4 Registers....611
8.8.3.4.1 COMP_REG_0....612
8.8.3.4.2 COMP_REG_1....613
8.8.3.4.3 MCU_WKUP_CTRL....613
8.8.3.4.4 AON_SLP_EVENT....614
8.8.3.4.5 PMU_COMP_GLITCH_REMOVE....614
8.8.3.5 Electrical Specifications....614
8.9 DMA....615
8.9.1 Introduction....615
8.9.2 Main Features....615
8.9.3 Block Diagram....615
8.9.4 Functional Description....617
8.9.4.1 Transfer Type Flow Control....617
8.9.4.2 Transfer Width....617
8.9.4.3 Source and Destination Address Increment....618
8.9.4.4 Channel Priority....618
8.9.4.5 DMA Transfers....618
8.9.4.6 Gather and Scatter....619
8.9.4.7 Multi-block DMA Transfer....622
8.9.4.7.1 Auto-reloading....622

8.9.4.7.2 Linked Lists (Block Chaining)..... 623
8.9.4.8 DMA Interrupt..... 624
8.9.4.9 DMA Hardware Interface Assignment..... 625
8.9.5 Registers..... 626
8.9.5.1 Channel_x_Registers..... 626
8.9.5.1.1 SARx..... 626
8.9.5.1.2 DARx..... 627
8.9.5.1.3 LLPx..... 627
8.9.5.1.4 CTLx..... 628
8.9.5.1.5 SSTATx..... 633
8.9.5.1.6 DSTATx..... 634
8.9.5.1.7 SSTATARx..... 634
8.9.5.1.8 DSTATARx..... 635
8.9.5.1.9 CFGx..... 635
8.9.5.1.10 SGRx..... 640
8.9.5.1.11 DSRx..... 640
8.9.5.2 Interrupt_Registers..... 641
8.9.5.2.1 INT_RSTAT_TC..... 641
8.9.5.2.2 INT_RSTAT_BTC..... 642
8.9.5.2.3 INT_RSTAT_STC..... 642
8.9.5.2.4 INT_RSTAT_DTC..... 643
8.9.5.2.5 INT_RSTAT_ERR..... 643
8.9.5.2.6 INT_STAT_TC..... 644
8.9.5.2.7 INT_STAT_BTC..... 644
8.9.5.2.8 INT_STAT_STC..... 645
8.9.5.2.9 INT_STAT_DTC..... 645
8.9.5.2.10 INT_STAT_ERR..... 646
8.9.5.2.11 INT_MASK_TC..... 646
8.9.5.2.12 INT_MASK_BTC..... 647
8.9.5.2.13 INT_MASK_STC..... 648
8.9.5.2.14 INT_MASK_DTC..... 649
8.9.5.2.15 INT_MASK_ERR..... 650
8.9.5.2.16 INT_CLR_TC..... 650
8.9.5.2.17 INT_CLR_BTC..... 651
8.9.5.2.18 INT_CLR_STC..... 651
8.9.5.2.19 INT_CLR_DTC..... 652
8.9.5.2.20 INT_CLR_ERR..... 652
8.9.5.2.21 INT_STAT_ET..... 653
8.9.5.3 Software_Handshake_Registers..... 654
8.9.5.3.1 REQ_SST..... 654
8.9.5.3.2 REQ_DST..... 654
8.9.5.3.3 REQ_SGL_ST..... 655

8.9.5.3.4 REQ_SGL_DT 656
8.9.5.3.5 REQ_LST_ST 656
8.9.5.3.6 REQ_LST_DT 657
8.9.5.4 Miscellaneous_Registers 658
8.9.5.4.1 CFG 658
8.9.5.4.2 CH_EN 658

9 Communication Subsystem 660
9.1 Supported Features 660
9.2 Transceiver 660
9.3 Digital Bluetooth LE Subsystem 660
9.4 Performance 661
9.4.1 General Performance 661
9.4.2 Receiver Performance 661
9.4.3 Transmitter Performance 665

10 Package Information 667
10.1 BGA83 667
10.2 QFN68 668

11 Ordering Codes 671

12 Tape and Reel Information 672

13 Glossary 674

14 Reference Documents 677

15 Legal and Contact Information 678

16 Revision History 679

1 GR5526 Overview

The Goodix GR5526 family is a single-mode, low-power Bluetooth 5.3 System-on-Chip (SoC). It can be configured as a Broadcaster, an Observer, a Central, and a Peripheral and supports the combination of all the above roles, making it an ideal choice for Internet of Things (IoT) and smart wearable devices. In addition, it supports Bluetooth Low Energy (Bluetooth LE) direction finding: angle of arrival (AoA) and angle of departure (AoD), as well as LE Audio, made possible by LE Isochronous Channels.

Based on ARM® Cortex®-M4F CPU core, the GR5526 series integrates Bluetooth 5.3 Protocol Stack, a 2.4 GHz RF transceiver, on-chip programmable Flash memory, RAM, multiple peripherals, enhanced I2C/UART port number for sensor applications, as well as expanded I/O functionality. GR5526 delivers a feature-rich display and graphics solution by providing the choice of graphics acceleration (GPU + DC) and internal/external system-in-package (SiP) pseudostatic RAM (PSRAM) to accommodate display while still leaving plenty of space for wearable schemes.

GR5526 series comes in two package choices: BGA83 and QFN68 (as shown in the table below), that can meet diverse application scenarios.

Table 1-1 GR5526 series

Features	GR5526VGBIP	GR5526VGBI	GR5526RGNIP	GR5526RGNI
CPU	Cortex®-M4F	Cortex®-M4F	Cortex®-M4F	Cortex®-M4F
RAM	512 KB	512 KB	512 KB	512 KB
SiP Flash	1 MB	1 MB	1 MB	1 MB
SiP PSRAM	8 MB	N/A	8 MB	N/A
GPU + DC	Supported	N/A	Supported	N/A
I/O number	50	50	48	48
Package (mm)	BGA83 (4.3 x 4.3 x 0.96)	BGA83 (4.3 x 4.3 x 0.96)	QFN68 (7.0 x 7.0 x 0.85)	QFN68 (7.0 x 7.0 x 0.85)

1.1 Features

- Bluetooth Low Energy 5.3 transceiver integrating Controller and Host layers
  - Supported data rates: 1 Mbps, 2 Mbps, and Long Range (500 kbps, 125 kbps)
  - TX power: -20 dBm to +7 dBm
  - -98 dBm sensitivity (in 1 Mbps mode)
  - -94 dBm sensitivity (in 2 Mbps mode)
  - -101 dBm sensitivity (in Long Range 500 kbps mode)
  - -104 dBm sensitivity (in Long Range 125 kbps mode)
  - TX current: 4.0 mA @ 0 dBm, 1 Mbps
  - RX current: 3.5 mA @ 1 Mbps
  - AoA/AoD, LE Isochronous Channels

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

- ARM® Cortex®-M4F 32-bit micro-processor with floating point support
  - Up to 96 MHz clock frequency
  - Built-in Memory Protection Unit (MPU) supporting eight programmable regions
  - Hardware Floating Point Unit (FPU)
  - Built-in Nested Vectored Interrupt Controller (NVIC)
  - Non-maskable Interrupt (NMI) input
  - Serial Wire Debug (SWD) with 16 breakpoints, two watchpoints, and a debug timestamp counter
  - 51 µA/MHz execution from Flash @ 3.3 V, 96 MHz
- On-chip memory
  - 512 KB data SRAM with retention capabilities
  - 8 KB cache SRAM with retention capabilities
  - Stack ROM (including boot ROM and Bluetooth LE Stack)
  - 1 MB internal QSPI Flash
  - 8 MB internal PSRAM (for GR5526VGBIP and GR5526RGNIP only)
- Digital peripherals
  - Two general-purpose DMA engines, each with six channels and up to 16 handshaking interfaces
  - USB 2.0 full speed (12 Mbps) controller with on-chip PHY and dedicated DMA controller
  - Internal Octal SPI DDR interfaces to support 8 MB internal PSRAM at up to 48 MHz (for GR5526VGBIP and GR5526RGNIP only)
- Analog peripherals
  - One 13-bit Sense ADC with a sampling rate of 1 Msps. It supports up to 8 external I/O channels and 3 internal signal channels.
  - Built-in temperature and voltage sensors
  - Low-power comparator, supporting wakeup from deep sleep mode
- Flexible serial peripherals
  - 6 x UART modules up to 4 Mbps, with all modules supporting flow control and IrDA
  - 6 x I2C modules for peripheral communication, up to 3.4 MHz
  - 1 x 8-bit/16-bit/32-bit SPI master interface and 1 x SPI slave interface for host communication
  - 2 x I2S interfaces (1 I2S master interface and 1 I2S slave interface)
  - PDM interface with hardware sampling rate converter
  - 1 x ISO7816 interface

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

- Display/Graphics
  - 2.5D GPU for graphics acceleration (for GR5526VGBIP and GR5526RGNIP only)
  - 1 x Dual-lane SPI (DSPI) interface for display, with Mobile Industry Processor Interface (MIPI) Display Bus Interface (DBI) Type-C support
  - 3 x Quad SPI (QSPI) interfaces, up to 48 MHz; supporting direct access via memory mapping when connecting with external NOR Flash
  - Display Controller (DC) module with MIPI DBI Type-C support and 2D graphics blending integrated (for GR5526VGBIP and GR5526RGNIP only)
- Security
  - Complete secure computing engine:
    - AES 128-bit/192-bit/256-bit symmetric encryption (ECB, CBC)
    - Hash-based Message Authentication Code (HMAC-SHA256)
    - Public key cryptography (PKC)
    - True random number generator (TRNG)
  - Comprehensive security operation mechanism:
    - Secure boot
    - Encrypted firmware running directly from Flash
    - eFuse for encrypted key storage
    - Differentiate application data key and firmware key, supporting one data key per each device/product
- I/O Peripherals
  - 50 I/O pins in total
    - 34 general-purpose I/O (GPIO) pins
    - 8 always-on I/O (AON IO) pins, supporting wakeup from deep sleep mode
    - 8 mixed signal I/O (MSIO) pins, configurable to be digital/analog signal interface
- Timer
  - Two general-purpose, 32-bit timer modules
  - A timer module composed of two programmable 32-bit or 16-bit down counters
  - An internal sleep timer that can be used to wake the device up from deep sleep mode
  - Two PWM modules with edge alignment mode and center alignment mode, each with 3 channels
  - 2 x real-time counters (RTC): 1 x Calendar, 1 x RTC
- Power management

- On-chip DC-DC to provide RF Analog voltage and supply core LDO
- On-chip I/O LDO to provide I/O voltage and supply external components, maximum I/O LDO drive strength: 30 mA
- Programmable thresholds for brownout detection (BOD)
- Supply voltage: 2.4 V–4.35 V
- I/O voltage: 1.8 V–3.6 V

- Low-power consumption modes
  - Sleep mode: 3.3 µA (Typical) at 3.3 V VBAT input with 128 KB SRAM retention on and LFXO_32K off; woken up by 8 sources of always-on domain
  - Ultra deep sleep mode: 2.4 µA (Typical); internal power (all SRAM included) and LFXO_32K removed from entire chip except always-on domain; woken up by Sleep Timer and AON GPIOs
  - OFF mode: 200 nA (Typical); nothing on except VBAT, and chip in reset mode
- Packages
  - BGA83: 4.3 x 4.3 x 0.96 (mm), 0.4 mm pitch
  - QFN68: 7.0 x 7.0 x 0.85 (mm), 0.35 mm pitch
- Operating temperature range: -40°C to +85°C

1.2 Block Diagram

The block diagram of GR5526 is shown in the figure below.

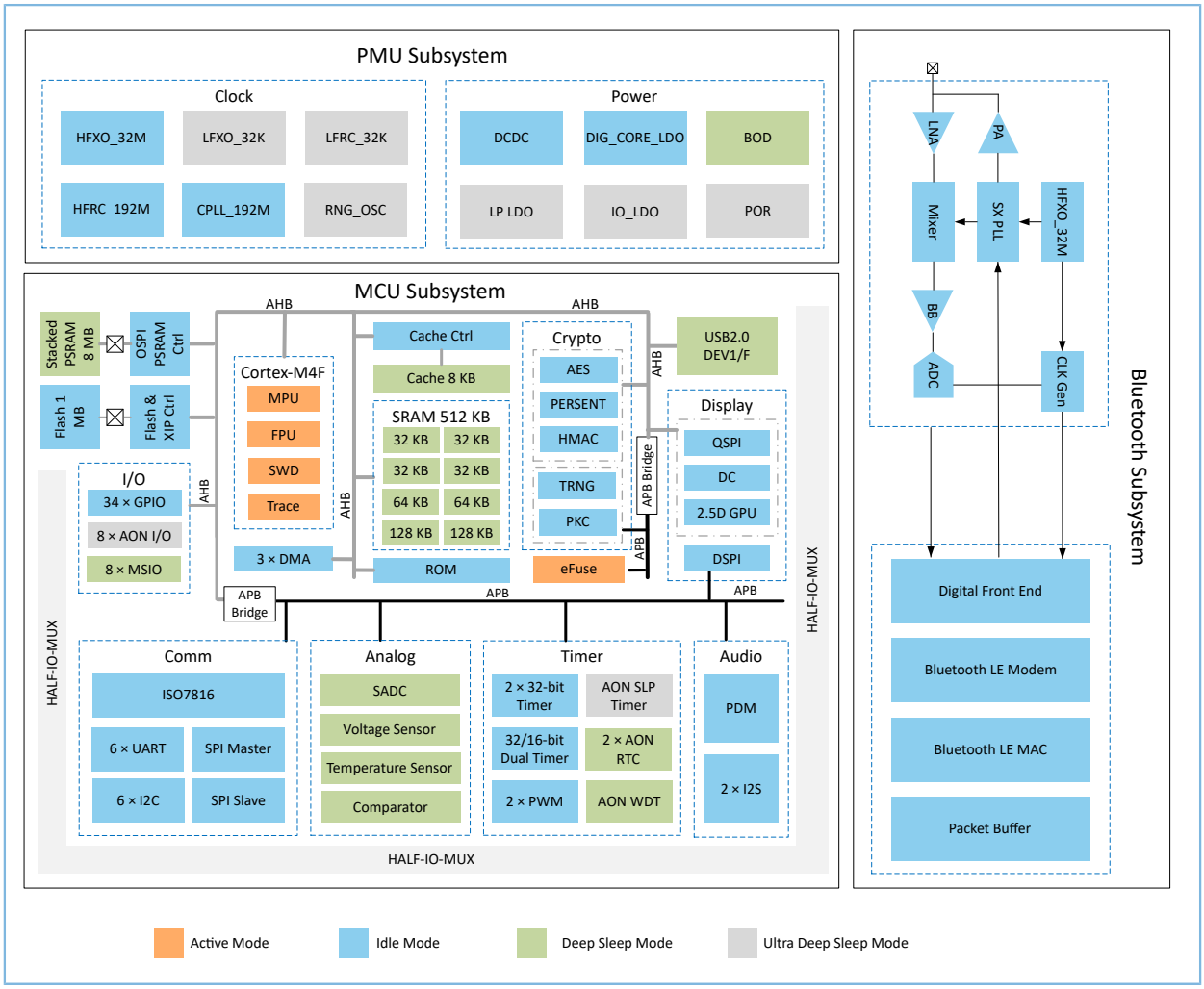

Figure 1-1 GR5526 block diagram

- Bluetooth subsystem
  - A 2.4 GHz transceiver and a digital communication core that supports Bluetooth 5.3
- MCU subsystem
  - An ARM® Cortex®-M4F with all the required memories and peripherals
  - Security cores that can be used for application security and secure boot implementation
  - GPU + DC to enhance processing capabilities
- Power management unit (PMU) subsystem
  - All the required power management modules to supply sufficient power for both internal modules and external peripherals
  - Modules required for ultra-low-power operation are in standby mode. HFRC_192M, RNG_OSC, LFRC_32K, wakeup GPIOs (Wake up), low-power comparator (LP Comp.), and power state controller (Power Sequencer) are used to control the state of different modules

1.3 Applications

The GR5526 SoC can be used in rich sets of applications. Example applications include:

- Advanced wearables
  - Sport bracelet
  - Smart watch
- Bluetooth HID devices
  - Voice remote control
  - Keyboard/Mouse
  - Gaming controller
  - Stylus pen
- IoT applications
  - Smart home
  - Electronic shelf label (ESL)
  - Beacon
  - Tire pressure monitoring system (TPMS)
  - Mesh applications
  - Asset tracking

2 Pinout

This chapter introduces GR5526 pinout available in BGA83 and QFN68 packages and provides detailed descriptions.

2.1 BGA83

The figure below shows the pins assignment (top view) of a GR5526 BGA83 package that is applicable to GR5526VGBIP and GR5526VGBI.

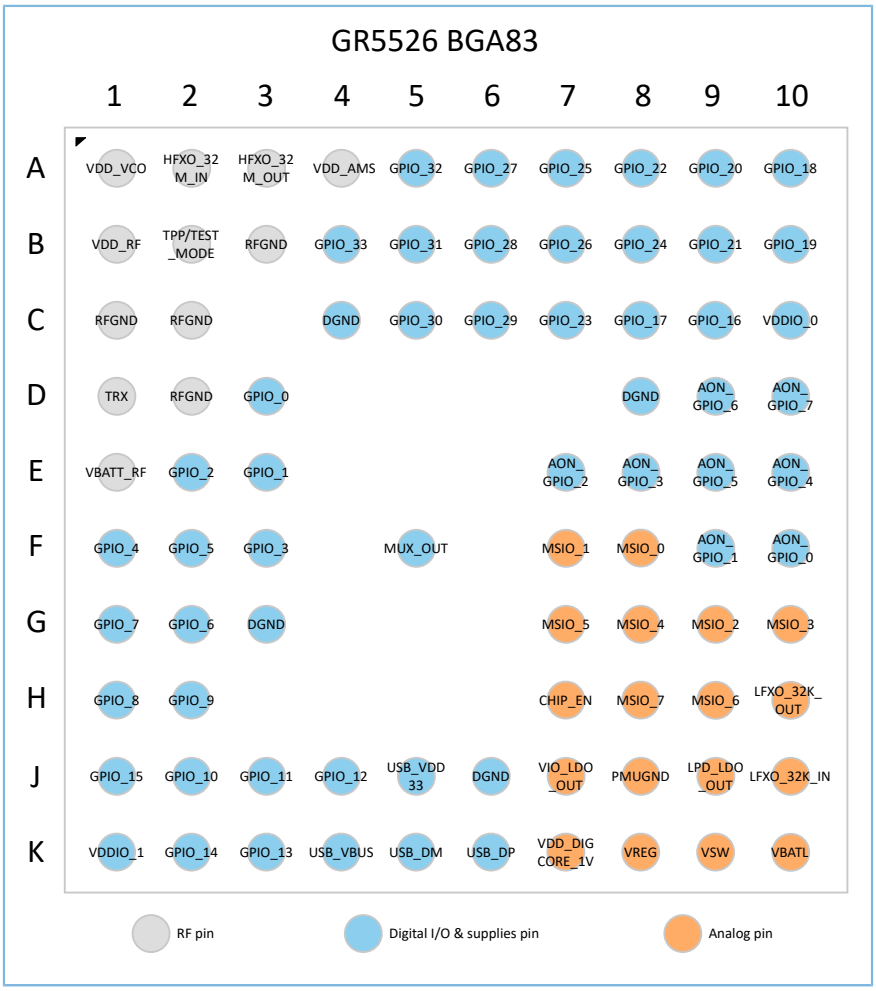

Figure 2-1 GR5526 BGA83 package pinout

The table below shows pin descriptions of a GR5526 BGA83 package.

Table 2-1 GR5526 BGA83 package pin descriptions

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
A1	VDD_VCO	Analog/RF Supply	Synthesizer VCO supply; connect to VREG	
A2	HFXO_32M_IN	Analog/RF	Input of inverting amplifier connected to 32 MHz crystal	
A3	HFXO_32M_OUT	Analog/RF	Output of inverting amplifier connected to 32 MHz crystal	
A4	VDD_AMS	Analog/RF Supply	AMS supply: 1.1 V; connect to VREG	

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
A5	GPIO_32	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
A6	GPIO_27	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
A7	GPIO_25	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
A8	GPIO_22	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
A9	GPIO_20	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
A10	GPIO_18	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
B1	VDD_RF	Analog/RF supply	RF supply: 1.1 V; connect to VREG	
B2	TPP/TEST_MODE	Analog/RF	Input pin, used to set test mode for factory test. • If TEST_MODE = 1, the chip is in test mode for factory test. • If TEST_MODE = 0, the chip is in normal operation mode.	
B3	RFGND	RF	RF ground	
B4	GPIO_33	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
B5	GPIO_31	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
B6	GPIO_28	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
B7	GPIO_26	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
B8	GPIO_24	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
B9	GPIO_21	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
B10	GPIO_19	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
C1	RFGND	RF	RF ground	
C2	RFGND	RF	RF ground	
C4	DGND	Digital Ground	Digital ground	
C5	GPIO_30	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
C6	GPIO_29	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
C7	GPIO_23	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
C8	GPIO_17	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
C9	GPIO_16	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
C10	VDDIO_0	Digital Supply	I/O supply voltage. Support external 1.8 V–3.3 V input voltage.	VDDIO0
D1	TRX	Analog/RF	RX input and TX output	
D2	RF_GND	RF	RF ground	
D3	GPIO_0	Digital I/O	General purpose I/O; default: SWD_CLK; pad drive level is 4 mA.	VDDIO1

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
D8	DGND	Digital Ground	Digital ground	
D9	AON_GPIO_6	Digital I/O	Always-on GPIO; pad drive level is 4 mA.	VDDIO0
D10	AON_GPIO_7	Digital I/O	Always-on GPIO; pad drive level is 4 mA.	VDDIO0
E1	VBATT_RF	Analog/RF Supply	Connected to VBATL	
E2	GPIO_2	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
E3	GPIO_1	Digital I/O	General purpose I/O; default: SWD_IO; pad drive level is 4 mA.	VDDIO1
E7	AON_GPIO_2	Digital I/O	Always-on GPIO; pad drive level is 4 mA.	VDDIO0
E8	AON_GPIO_3	Digital I/O	Always-on GPIO; pad drive level is 4 mA.	VDDIO0
E9	AON_GPIO_5	Digital I/O	Always-on GPIO; pad drive level is 4 mA.	VDDIO0
E10	AON_GPIO_4	Digital I/O	Always-on GPIO; pad drive level is 4 mA.	VDDIO0
F1	GPIO_4	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
F2	GPIO_5	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
F3	GPIO_3	Digital I/O	General purpose I/O, configurable to be an SWO interface; pad drive level is 4 mA.	VDDIO1
F5	MUX_OUT	PMU	-	
F7	MSIO_1	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); pad drive level is 2 mA.	VBATL
F8	MSIO_0	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); pad drive level is 2 mA.	VBATL
F9	AON_GPIO_1	Digital I/O	Always-on GPIO; pad drive level is 4 mA.	VDDIO0
F10	AON_GPIO_0	Digital I/O	Always-on GPIO; pad drive level is 4 mA.	VDDIO0
G1	GPIO_7	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
G2	GPIO_6	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
G3	DGND	Digital Ground	Digital ground	
G7	MSIO_5	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); pad drive level is 2 mA.	VBATL
G8	MSIO_4	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); pad drive level is 2 mA.	VBATL
G9	MSIO_2	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); pad drive level is 2 mA.	VBATL
G10	MSIO_3	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); pad drive level is 2 mA.	VBATL
H1	GPIO_8	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
H2	GPIO_9	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
H7	CHIP_EN	Mixed Signal IN	Master Enable for chip reset pin The high level of CHIP_EN equals VBATL.	
H8	MSIO_7	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); pad drive level is 2 mA.	VBATL
H9	MSIO_6	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); pad drive level is 2 mA.	VBATL
H10	LFXO_32K_OUT	PMU	Output of inverting amplifier connected to 32.768 kHz crystal	
J1	GPIO_15	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
J2	GPIO_10	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
J3	GPIO_11	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
J4	GPIO_12	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
J5	USB_VDD33	USB	Internal output voltage: USB 3.3 V	
J6	DGND	Digital Ground	Digital ground	
J7	VIO_LDO_OUT	PMU	Output of on-chip I/O supply regulator	
J8	PMUGND	PMU	DC/DC converter supply and general battery GND	
J9	LPD_LDO_OUT	PMU	Output of low power domain LDO	
J10	LFXO_32K_IN	PMU	Input of inverting amplifier connected to 32.768 kHz crystal	
K1	VDDIO_1	Digital Supply	I/O supply voltage. Support external 1.8 V–3.3 V input voltage.	VDDIO1
K2	GPIO_14	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
K3	GPIO_13	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
K4	USB_VBUS	USB	Input voltage: USB 5 V	
K5	USB_DM	USB	USB D- Pin	
K6	USB_DP	USB	USB D+ Pin	
K7	VDD_DIGCORE_1V	PMU	On-chip LDO output for digital core	
K8	VREG	PMU	Feedback pin of switch regulator	
K9	VSW	PMU	DC/DC converter switching node	
K10	VBATL	PMU	Power supply: 2.4 V to 4.35 V	

2.2 QFN68

The figure below shows the pins assignment (top view) of a GR5526 QFN68 package that is applicable to GR5526RGNIP and GR5526RGNI.

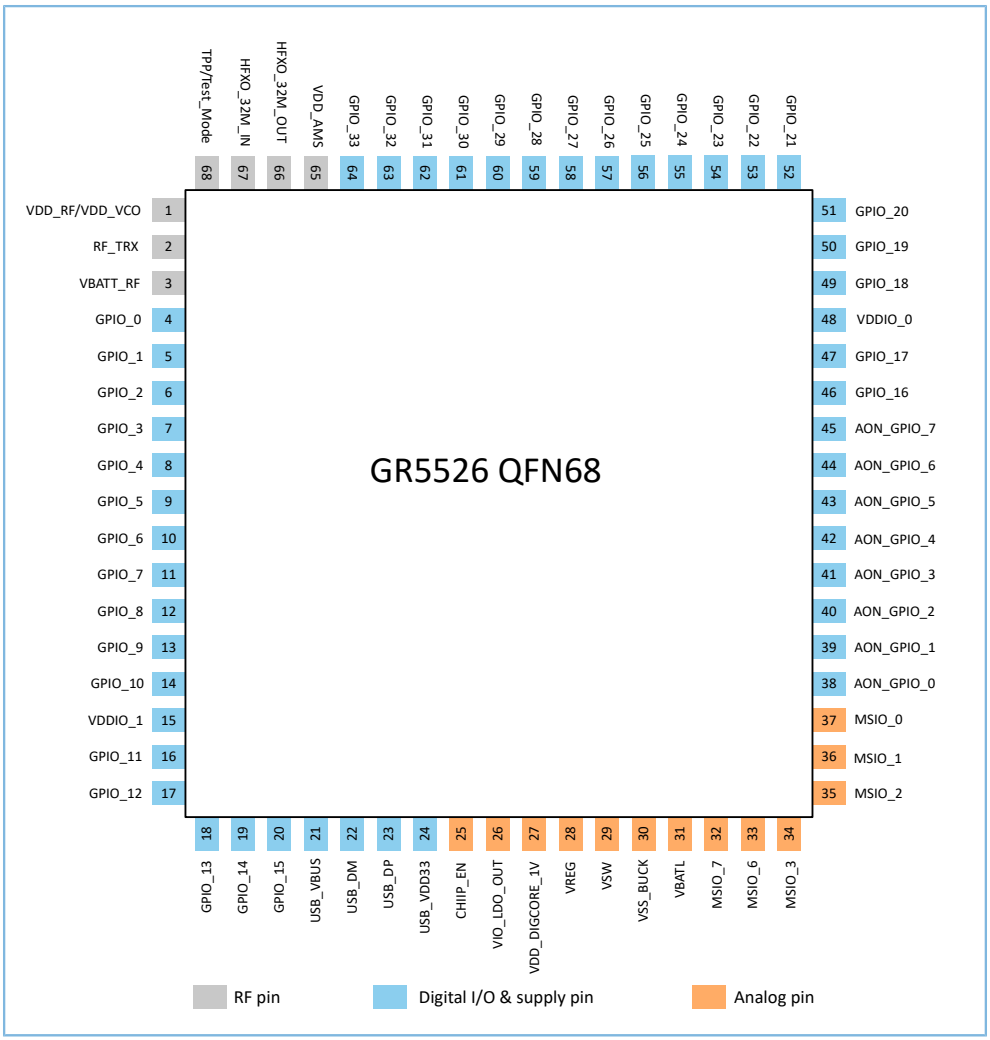

Figure 2-2 GR5526 QFN68 package pinout

The table below shows pin descriptions of a GR5526 QFN68 package.

Table 2-2 GR5526 QFN68 package pin descriptions

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
1	VDD_VCO/VDD_RF	Analog/RF supply	Synthesizer VCO supply/RF supply: 1.1 V; connect to VREG	
2	RF_TRX	Analog/RF	RX input and TX output	
3	VBATT_RF	Analog/RF Supply	Connected to VBATL	
4	GPIO_0	Digital I/O	General purpose I/O; default: SWD_CLK; pad drive level is 4 mA.	VDDIO1

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
5	GPIO_1	Digital I/O	General purpose I/O; default: SWD_IO; pad drive level is 4 mA.	VDDIO1
6	GPIO_2	Digital I/O	General purpose I/O, configurable to be an SWO interface; pad drive level is 4 mA.	VDDIO1
7	GPIO_3	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
8	GPIO_4	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
9	GPIO_5	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
10	GPIO_6	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
11	GPIO_7	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
12	GPIO_8	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
13	GPIO_9	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
14	GPIO_10	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
15	VDDIO_1	Digital I/O supply	Digital I/O supply input. Support external 1. 8 V–3.3 V input voltage.	VDDIO1
16	GPIO_11	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
17	GPIO_12	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
18	GPIO_13	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
19	GPIO_14	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
20	GPIO_15	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO1
21	USB_VBUS	USB	Input voltage: USB 5 V	
22	USB_DM	USB	USB D- pin	
23	USB_DP	USB	USB D+ pin	
24	USB_VDD33	USB	Internal output voltage: USB 3.3 V	
25	CHIP_EN	Mixed Signal IN	Master Enable for chip reset pin. The high value of CHIP_EN equals VBATL.	
26	VIO_LDO_OUT	PMU	Output of on-chip I/O supply regulator	
27	VDD_DIGCORE_1V	Analog/PMU	On-chip LDO output for digital core	
28	VREG	Analog/PMU	Feedback pin of switch regulator	
29	VSW	Analog/PMU	DC/DC converter switching node	
30	VSS_BUCK	Analog/PMU	DC/DC converter supply and general battery GND	
31	VBATL	Analog/PMU	Power supply: 2.4 V to 4.35 V	
32	MSIO_7	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); multiplexed by LFXO_32K_IN; pad drive level is 2 mA.	VBATL

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
33	MSIO_6	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); multiplexed by LFXO_32K_OUT; pad drive level is 2 mA.	VBATL
34	MSIO_3	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); pad drive level is 2 mA.	VBATL
35	MSIO_2	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); pad drive level is 2 mA.	VBATL
36	MSIO_1	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); pad drive level is 2 mA.	VBATL
37	MSIO_0	Mixed Signal I/O	Configurable to be a GPIO mixed signal (SNSADC interface); pad drive level is 2 mA.	VBATL
38	AON_GPIO_0	Digital I/O	Always-on general purpose I/O; pad drive level is 4 mA.	VDDIO0
39	AON_GPIO_1	Digital I/O	Always-on general purpose I/O; pad drive level is 4 mA.	VDDIO0
40	AON_GPIO_2	Digital I/O	Always-on general purpose I/O; pad drive level is 4 mA.	VDDIO0
41	AON_GPIO_3	Digital I/O	Always-on general purpose I/O; pad drive level is 4 mA.	VDDIO0
42	AON_GPIO_4	Digital I/O	Always-on general purpose I/O; pad drive level is 4 mA.	VDDIO0
43	AON_GPIO_5	Digital I/O	Always-on general purpose I/O; pad drive level is 4 mA.	VDDIO0
44	AON_GPIO_6	Digital I/O	Always-on general purpose I/O; pad drive level is 4 mA.	VDDIO0
45	AON_GPIO_7	Digital I/O	Always-on general purpose I/O; pad drive level is 4 mA.	VDDIO0
46	GPIO_16	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
47	GPIO_17	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
48	VDDIO_0	Digital I/O supply	Digital I/O supply input. Support external 1.8 V–3.3 V input voltage.	VDDIO0
49	GPIO_18	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
50	GPIO_19	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
51	GPIO_20	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
52	GPIO_21	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
53	GPIO_22	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
54	GPIO_23	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
55	GPIO_24	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
56	GPIO_25	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
57	GPIO_26	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
58	GPIO_27	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
59	GPIO_28	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
60	GPIO_29	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0

Pin #	Pin Name	Pin Type	Description/Default Function	Voltage Domain
61	GPIO_30	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
62	GPIO_31	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
63	GPIO_32	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
64	GPIO_33	Digital I/O	General purpose I/O; pad drive level is 4 mA.	VDDIO0
65	VDD_AMS	Analog/RF	AMS supply: 1.1 V. Connect to VREG.	
66	HFXO_32M_OUT	Analog/RF	Output of inverting amplifier connected to 32 MHz crystal	
67	HFXO_32M_IN	Analog/RF	Input of inverting amplifier connected to 32 MHz crystal	
68	TPP/TEST_MODE	Analog/RF	Input pin, used to set test mode for factory test. • If TEST_MODE = 1, the chip is in test mode for factory test. • If TEST_MODE = 0, the chip is in normal operation mode.	

2.3 Pin Properties

Table 2-3 I/O properties

Index	Pin Name	Default Status After POR	Pullup/ Pulldown Enable After POR	Pullup/ Pulldown Selection After POR	Pin Interrupt	Wakeup Chip	Sleep Retention
1	AON_GPIO_0	L	Y	PD	Y	Y	Y
2	AON_GPIO_1	L	Y	PD	Y	Y	Y
3	AON_GPIO_2	L	Y	PD	Y	Y	Y
4	AON_GPIO_3	L	Y	PD	Y	Y	Y
7	AON_GPIO_4	L	Y	PD	Y	Y	Y
8	AON_GPIO_5	L	Y	PD	Y	Y	Y
5	AON_GPIO_6	L	Y	PD	Y	Y	Y
6	AON_GPIO_7	L	Y	PD	Y	Y	Y
9	GPIO_0	L	Y	PD	Y	N	Y
10	GPIO_1	L	Y	PD	Y	N	Y
11	GPIO_2	L	Y	PD	Y	N	Y
12	GPIO_3	L	Y	PD	Y	N	Y
13	GPIO_4	L	Y	PD	Y	N	Y
14	GPIO_5	L	Y	PD	Y	N	Y
15	GPIO_6	L	Y	PD	Y	N	Y

Index	Pin Name	Default Status After POR	Pullup/ Pulldown Enable After POR	Pullup/ Pulldown Selection After POR	Pin Interrupt	Wakeup Chip	Sleep Retention
16	GPIO_7	L	Y	PD	Y	N	Y
17	GPIO_8	L	Y	PD	Y	N	Y
18	GPIO_9	L	Y	PD	Y	N	Y
19	GPIO_10	L	Y	PD	Y	N	N
20	GPIO_11	L	Y	PD	Y	N	N
21	GPIO_12	L	Y	PD	Y	N	N
22	GPIO_13	L	Y	PD	Y	N	N
23	GPIO_14	L	Y	PD	Y	N	N
24	GPIO_15	L	Y	PD	Y	N	N
25	GPIO_16	L	Y	PD	Y	N	N
26	GPIO_17	L	Y	PD	Y	N	N
27	GPIO_18	L	Y	PD	Y	N	N
28	GPIO_19	L	Y	PD	Y	N	N
29	GPIO_20	L	Y	PD	Y	N	N
30	GPIO_21	L	Y	PD	Y	N	N
31	GPIO_22	L	Y	PD	Y	N	N
32	GPIO_23	L	Y	PD	Y	N	N
33	GPIO_24	L	Y	PD	Y	N	N
34	GPIO_25	L	Y	PD	Y	N	N
35	GPIO_26	L	Y	PD	Y	N	Y
36	GPIO_27	L	Y	PD	Y	N	Y
37	GPIO_28	L	Y	PD	Y	N	Y
38	GPIO_29	L	Y	PD	Y	N	Y
39	GPIO_30	L	Y	PD	Y	N	Y
40	GPIO_31	L	Y	PD	Y	N	Y
41	GPIO_32	L	Y	PD	Y	N	N
42	GPIO_33	L	Y	PD	Y	N	N
43	MSIO_0	L	Y	PD	N	N	Y
44	MSIO_1	L	Y	PD	N	N	Y
45	MSIO_2	L	Y	PD	N	N	Y
46	MSIO_3	L	Y	PD	N	N	Y

Index	Pin Name	Default Status After POR	Pullup/ Pulldown Enable After POR	Pullup/ Pulldown Selection After POR	Pin Interrupt	Wakeup Chip	Sleep Retention
47	MSIO_4	L	Y	PD	N	N	Y
48	MSIO_5	L	Y	PD	N	N	Y
49	MSIO_6	L	Y	PD	N	N	Y
50	MSIO_7	L	Y	PD	N	N	Y

Table 2-4 Abbreviations related to pin properties

Properties	Abbreviation	Description
Default status after POR	Hi-Z	High impedance
	H	High level
	L	Low level
Pullup/Pulldown enable after POR	Y	Enabled
	N	Disabled
Pullup/Pulldown selection after POR	PU	Pullup
	PD	Pulldown
Pin interrupt	N	No
	Y	Yes

3 MCU Core

3.1 Functional Overview

The GR5526 core includes on-chip OSPI PSRAM, built-in XIP FLASH, and high-performance 32-bit ARM® Cortex®-M4F processor. In addition, the GR5526 core has a series of high-speed peripherals such as digital communication, analog interface, and IO. For details, see Chapter 8 Peripherals.

The Cortex®-M4F is built on a high-performance processor core with a 3-stage pipeline Harvard architecture, making it ideal for demanding embedded applications. The processor delivers exceptional power efficiency through an efficient instruction set and extensively optimized design, providing high-end processing hardware including optional IEEE754-compliant single-precision floating-point computation, a range of single-cycle and SIMD multiplication and multiply-with-accumulate capabilities, saturating arithmetic and dedicated hardware division.

To facilitate the design of cost-sensitive devices, the Cortex®-M4F processor implements tightly-coupled system components that reduce processor area while significantly improving interrupt handling and system debug capabilities. The Cortex®-M4F processor implements a version of the Thumb® instruction set based on Thumb-2 technology, ensuring high code density and reduced program memory requirements. The Cortex®-M4F instruction set provides the exceptional performance expected of a modern 32-bit architecture, with the high code density of 8-bit and 16-bit microcontrollers.

The Cortex®-M4F processor closely integrates a configurable Nested Vectored Interrupt Controller (NVIC), to deliver industry-leading interrupt performance. The NVIC includes a Non Maskable Interrupt (NMI) that can provide up to 256 interrupt priority levels. The tight integration of the processor core and NVIC provides fast execution of Interrupt Service Routines (ISRs), dramatically reducing the interrupt latency. This is achieved through the hardware stacking of registers, and the ability to suspend load-multiple and store-multiple operations. Interrupt handlers do not require wrapping in assembler code, removing any code overhead from the ISRs. A tail-chain optimization also significantly reduces the overhead when switching from one ISR to another.

The Cortex®-M4F processors include many debug features that allow developers to analyze design problems easily. Besides standard design features which developers can find in most microcontrollers like halting and single stepping, developers can also generate a trace to capture program flow, data changes, profiling information, and more.

The ARM® Cortex®-M4F processor offers a number of CPU options and support modules implemented on the device.

Table 3-1 Support modules

No.	Option/Module	Description	Implemented
1	NVIC	Nested vector interrupt controller	66 vectors
2	PRIORITIES	Priority bits	8 bits
3	Endianness	Memory system endianness	Little endian
4	Bit-banding	Bit banded memory	Yes
5	DWT	Data watchpoint and trace	Yes
6	SysTick	System tick timer	Yes

No.	Option/Module	Description	Implemented
7	FPU	Floating-point unit	Yes
8	MPU	Memory protection unit	8 regions
9	DAP	Debug access port	Yes
10	ETM	Embedded trace macrocell	No
11	ITM	Instrumentation trace macrocell	Yes
12	TPIU	Trace port interface unit	Yes
13	ETB	Embedded trace buffer	No
14	FPB	Flash patch and breakpoint unit	16 addresses

Benefiting from the efficient design of each module, GR5526 can easily reach a high level of operating efficiency. The CoreMark score is shown in the table below.

Table 3-2 MCU CoreMark score

Symbol	Description	Min.	Typ.	Max.	Unit
CM_{FLASH}	CoreMark, running from Flash, 96 MHz, direct map cache		316		CoreMark
$CM_{FLASH/MHz}$	CoreMark per MHz, running from Flash, 96 MHz, direct map cache		3.3		CoreMark/MHz
CM_{FLASH}	CoreMark, running from Flash, 96 MHz, 4-way associative cache		261		CoreMark
$CM_{FLASH/MHz}$	CoreMark per MHz, running from Flash, 96 MHz, 4-way associative cache		2.7		CoreMark/MHz

3.2 Interrupt

In addition to supporting 16 system exceptions within the ARM® Cortex®-M4F, the GR5526 series also supports 66 external interrupts with 256 programmable priority levels. In some cases, a single peripheral may be able to generate multiple different interrupts (such as BOD/BLE). Most interrupts are connected directly to the NVIC, but all wakeup sources on the always-on domain are connected to the Power State Controller (PSC) outside of the ARM® Cortex®-M4F, allowing the interrupt sources to wake up the ARM® Cortex®-M4F core from the deep sleep mode. In most cases, developers do not need to change the interrupt source.

The Cortex®-M4F allows developers to assign various interrupts to different priority levels based on the application requirements. In order to maintain the stability of Bluetooth connection, it is recommended that the priority of Bluetooth interrupt should be configured to be the highest. In most cases, developers do not need to change priority levels, and the GR5526 SoCs have been configured with default priorities to meet most application requirements.

One additional feature of the Cortex® -M4F interrupt architecture is the ability to relocate the vector table to different addresses. The software can move the vector table into SRAM and reassign the interrupt service routine entry addresses on demand, but it should be noted that the remapping start address of the interrupt vector table should be aligned with **0x200**. For example, the vector table base address should be set as **0x00000000**, **0x00000200**, or **0x00000400**.

All non-programmable and programmable interrupts including those that can wake the MCU up from deep sleep mode are shown in Table 3-3 and Table 3-4. The width of interrupt priority control register is 8 bits, in which bit7 to bit4 are used for preemption priority and bit3 to bit0 are used for subpriority.

Table 3-3 Non-programmable interrupts

No.	Name	Default Priority	Waking up MCU	Description
1	Reset_IRQ	-3		Reset exception
2	NMI_IRQn	-2		Non-maskable interrupt
3	HardFault_IRQ	-1		All classes of fault. This interrupt occurs when the corresponding fault handler cannot be activated because it is currently disabled or masked by exception masking.

Table 3-4 Programmable interrupts

No.	Name	Default Preemption Priority	Default Subpriority	Waking up MCU	Description
4	MemManage_IRQ	-	-		Memory management fault; caused by MPU violation or invalid accesses
5	BusFault_IRQ	-	-		Error response received from the bus system
6	UsageFault_IRQ	-	-		Usage fault; typical causes are invalid instructions or invalid state transition attempts.
7–10	Reserved	-	-		Unused
11	SVCall_IRQ	0	0		Supervisor call via SVC instruction
12	DebugMonitor_IRQ	-	-		Debug monitor – for software-based debugging
13	Reserved	-	-		Unused
14	PendSV_IRQ	15	15		Pendable request for system service
15	SysTick_IRQ	15	15		System tick timer interrupt
16	Reserved	-	-		Reserved
17	BLE_SDK_IRQ	15	15		BLE software development kit (SDK) schedule interrupt
18	BLE_IRQ	2	0	√	BLE interrupt
19	DMA0_IRQ	6	0		DMA0 interrupt

No.	Name	Default Preemption Priority	Default Subpriority	Waking up MCU	Description
20	SPI_M_IRQ	8	0		SPI master interrupt
21	SPI_S_IRQ	8	0		SPI slave interrupt
22	EXT0_IRQ	8	0		GPIO_0 interrupt
23	EXT1_IRQ	8	0		GPIO_1 interrupt
24	TIMER0_IRQ	12	0		TIMER0 interrupt
25	TIMER1_IRQ	12	0		TIMER1 interrupt
26	DUAL_TIMER_IRQ	12	0		DUAL_TIMER interrupt
27	QSPI0_IRQ	8	0		QSPI0 interrupt
28	UART0_IRQ	10	0		UART0 interrupt
29	UART1_IRQ	10	0		UART1 interrupt
30	I2C0_IRQ	8	0		I2C0 interrupt
31	I2C1_IRQ	8	0		I2C1 interrupt
32	AES_IRQ	8	0		AES interrupt
33	HMAC_IRQ	8	0		HMAC interrupt
34	EXT2_IRQ	8	0		GPIO_2 interrupt
35	RNG_IRQ	8	0		TRNG interrupt
36	BOD_ASSERT_IRQ	8	0		Brown-out detect interrupt
37	PKC_IRQ	8	0		PKC interrupt
38	XQSPI_IRQ	8	0		XQSPI interrupt
39	QSPI1_IRQ	8	0		QSPI1 interrupt
40	OSPI_IRQ	8	0		OSPI interrupt
41	BLESLP_IRQ	2	0	√	BLE timer done interrupt
42	SLPTIMER_IRQ	12	0	√	Sleep timer done interrupt
43	AON_EXT_IRQ	8	0	√	AON GPIO interrupt
44	AON_WDT_IRQ	8	0	√	AON watch dog timer interrupt
45	I2S_M_IRQ	8	0		I2S master interrupt
46	I2S_S_IRQ	8	0		I2S slave interrupt
47	ISO7816_IRQ	8	0		SIM card interrupt
48	PRESENT_IRQ	8	0		Present done interrupt
49	RTC0_IRQ	8	0	√	Real-timer counter0 interrupt
50	COMM_CORE_IRQ	8	0		Communication core
51	DMA1_IRQ	6	0		DMA1 interrupt

No.	Name	Default Preemption Priority	Default Subpriority	Waking up MCU	Description
52	DMA2_IRQ	6	0		DMA2 interrupt
53	DSPI_IRQ	8	0		DSPI interrupt
54	AON_IRQ	8	0		AON domain interrupt
55	PDM_IRQ	8	0		PDM interrupt
56	VTTBL_IRQ	8	0		VT table interrupt
57	CTE_FULL_IRQ	8	0		CTE full interrupt
58	USB_IRQ	8	0		USB interrupt
59	TSI_DC_IRQ	8	0		Display control interrupt
60	BOD_DEASSERT_IRQ	8	0	√	Brown out detect disappear interrupt
61	COMP_IRQ	8	0	√	Comparator interrupt
62	USB_ATTACH_IRQ	8	0	√	USB attach interrupt
63	USB_DETACH_IRQ	8	0	√	USB detach interrupt
64	RTC1_IRQ	8	0	√	Real-timer counter1 interrupt
65	CPLL_DRIFT_IRQ	8	0		CPLL drift interrupt
66	CLK_CALIB_IRQ	8	0		Clock calibration0/1 interrupt
67	TSI_GPU_IRQ	8	0		GPU interrupt
68	UART2_IRQ	10	0		UART2 interrupt
69	UART3_IRQ	10	0		UART3 interrupt
70	UART4_IRQ	10	0		UART4 interrupt
71	UART5_IRQ	10	0		UART5 interrupt
72	BLE_PWR_ON_IRQ	2	0		BLE sequencer power on done interrupt
73	BLE_PWR_DN_IRQ	8	0		BLE sequencer power off done interrupt
74	I2C2_IRQ	8	0		I2C2 interrupt
75	I2C3_IRQ	8	0		I2C3 interrupt
76	I2C4_IRQ	8	0		I2C4 interrupt
77	I2C5_IRQ	8	0		I2C5 interrupt
78	XO_BIAS_SW_CHG_IRQ	8	0		Crystal oscillator (XO) bias switch changed interrupt
79	AVS_ERR_IRQ	8	0		AVS control error interrupt
80	TSI_GPU_ERR_IRQ	8	0		GPU error interrupt
81	QSPI2_IRQ	8	0		QSPI2 interrupt

3.3 Memory Map

Referring to the Cortex®-M processor storage architecture, a GR5526 SoC divides its memory into ROM, Flash, data memory, peripheral memory, and memory for debugging components. All the mapping spaces of the 4 GB SoC memory are shown in Figure 3-1.

- The contents of the ROM area have been preprogrammed and cannot be modified by developers. This area mainly stores the startup process and code related to BLE protocol stack.
- The RAM Alias1 area is used to store special-purpose code segments that are generally used for Flash operations or to speed up code fetching. Due to the characteristics of the Cortex®-M4F bus architecture, there are independent instruction bus (I-CODE) and data bus (D-CODE) in the address space of **0x0000 0000**–**0x1FFF FFFF**. The two buses can be used to fetch instructions and data in parallel, and the address space of **0x2000 0000**–**0xDFFF FFFF** has only one system bus (System) where fetching instructions and data can only be performed serially. Therefore, it is more efficient to put code segments and constants in the area of **0x0000 0000**–**0x1FFF FFFF**.
- The ExFlash area is generally used to store user code. ExFlash area and ExFlash Alias area share the same physical address. In non-encryption mode, the code data in the ExFlash area can also be read by the corresponding area of ExFlash Alias. In encryption mode, firmware key is required to support reading the content of ExFlash, and data key is required to support reading the content of ExFlash Alias.
- The RAM area is generally used to store data and code copied from Flash. In the ARM® Cortex®-M4F architecture, the bit-band operations in the **0x2000 0000**–**0x200F FFFF** and **0x4000 0000**–**0x400F FFFF** are conducive to the atomic characteristics of data operations.
- QSPI0, QSPI1, QSPI2, and OSPI all support XIP mode. In this mode, the data space of Flash or PRSAM can be mapped to the memory, which allows for direct operations on the memory address. For external PSRAM solutions, the external PSRAM mounted on QSPI M1 XIP Alias1 can be combined with the memory of **0x2000 0000**–**0x2007 FFFF** to form a continuous SRAM address space. For internal PSRAM solutions, the internal PSRAM mounted on OSPI XIP Alias can be combined with the **0x3000 0000** – **0x3007 FFFF** memory area to form a continuous SRAM address space.

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

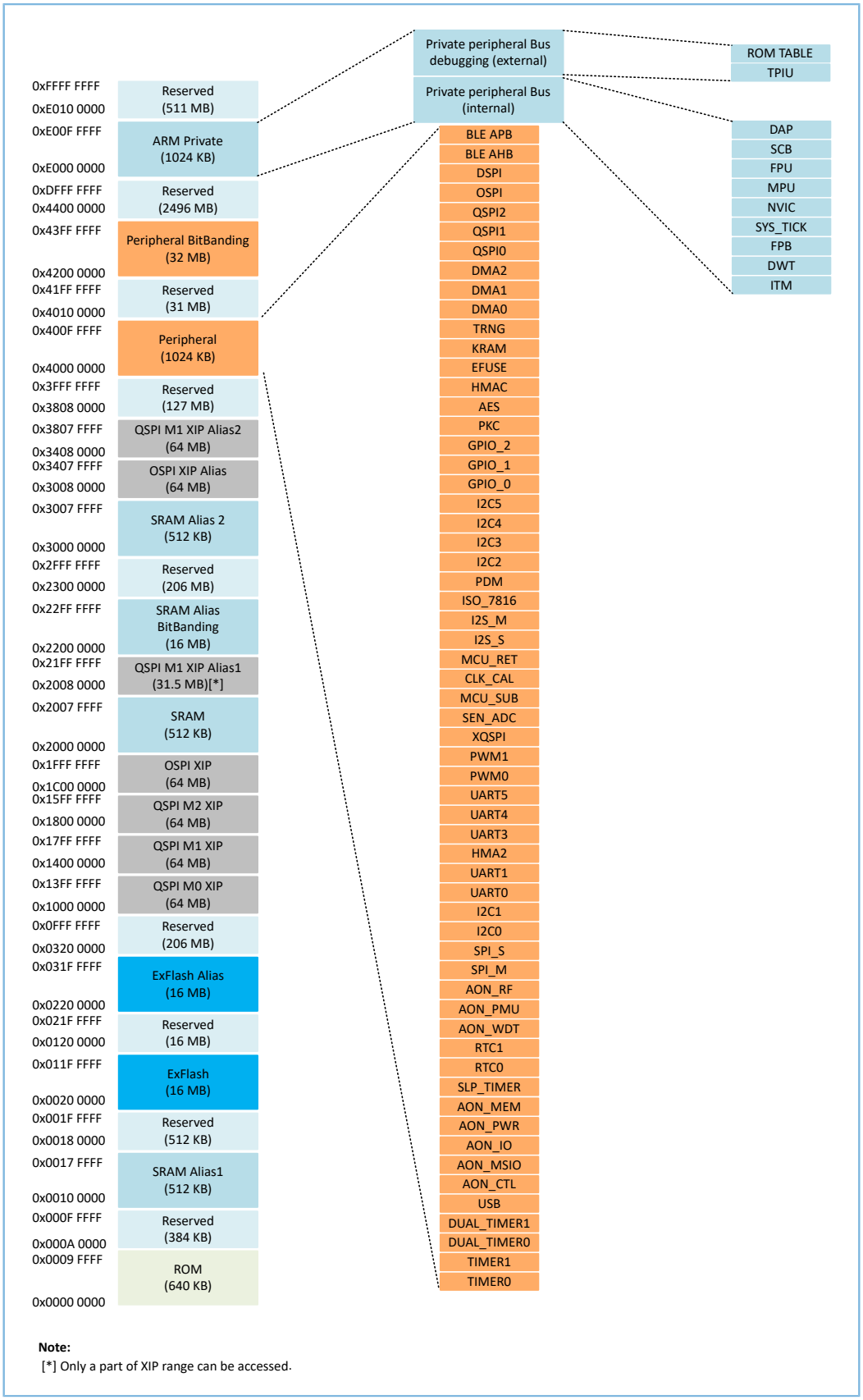

Figure 3-1 GR5526 memory map

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

3.4 System Buses

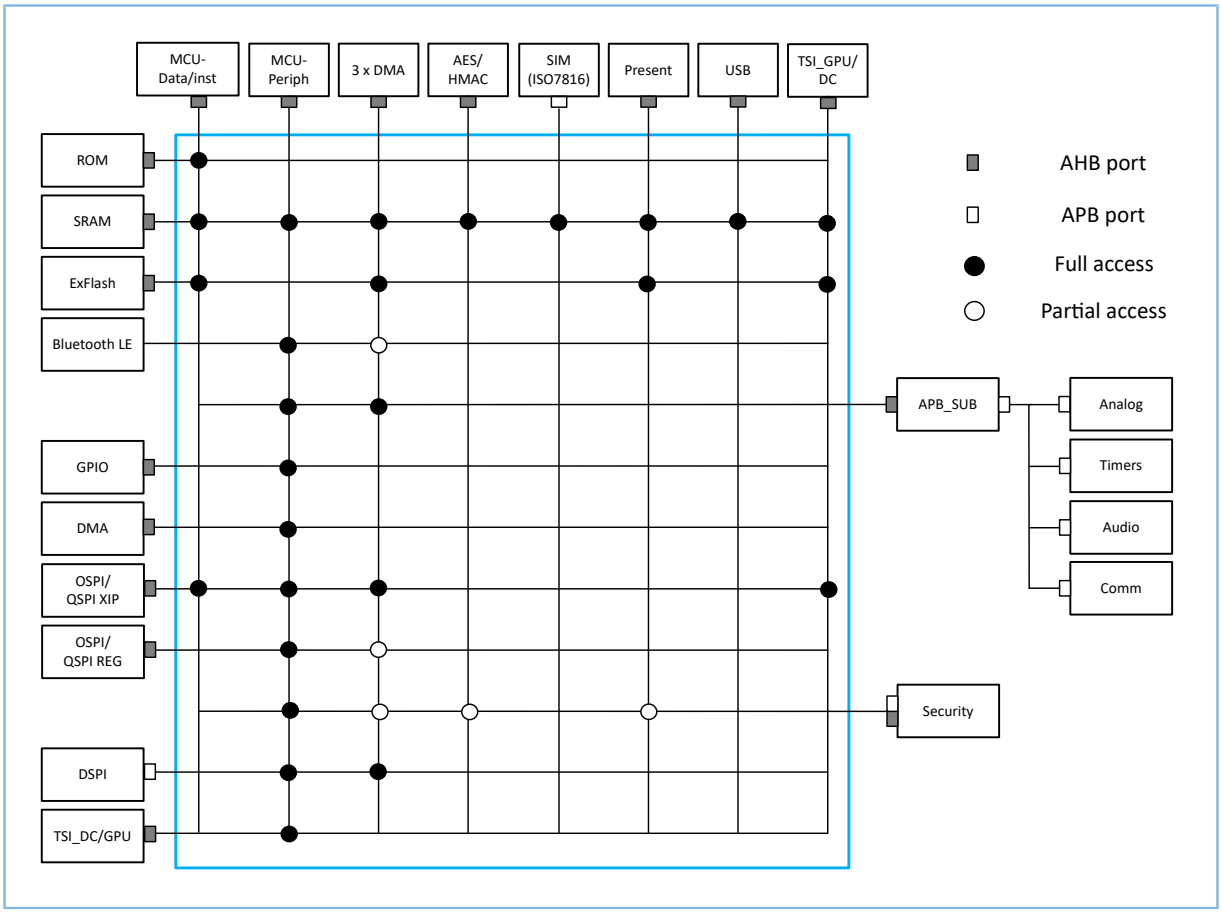

Figure 3-2 GR5526 system bus

The ARM® Cortex®-M4F processors provide generic bus interfaces based on Advanced Microcontroller Bus Architecture (AMBA). The AMBA specification supports several bus protocols for communication with memories and peripherals.

- I-Code Bus interface: Primarily for program memory; fetches instructions from the 'Code' memory space (0x0 to 0x1FFFFFFF).
- D-Code Bus interface: Primarily for program memory; supports data and debug accesses to the 'Code' memory space (0x0 to 0x1FFFFFFF).
- System Bus interface: Primarily for RAM and peripherals; accesses to the SRAM and other peripherals of the MCU.

Among them, I-CODE and D-CODE are independent from the system bus, which can realize parallel data access and value acquisition, effectively improving performance. The GR5526 MCU maps the available SRAM onto an address space within the 'Code' memory space. This gives the application the opportunity to perform tasks efficiently because the system can quickly finish the task and go to sleep mode.

AMBA supports a variety of bus protocols. For GR5526, the main bus interface uses the AHB Lite protocol. The infrequently accessed peripherals of the GR5526 MCU are located on an AMBA APB bus. There is a bridge which

translates the accesses from the System AHB to the APB. Access to these peripherals will inject a single wait-state on the AHB during any access cycle. The system bus diagram of GR5526 is shown in Figure 3-2.

At the same time, the AHB bus has matrix ports, which can support DMA, CPU, and USB to access different memories and peripherals at the same time, beneficial to improve the performance of data access. When multiple masters access the same memory or peripherals, GR5526 uses a round-robin scheduling to configure the priorities of multiple masters. After each response to the master's request, the priority will be determined according to the order of access. This method is more flexible than fixed priority.

3.5 Debug Interfaces

The GR5526 SoCs are built around a Cortex®-M4F with FPU core which contains hardware extensions for advanced debugging features. The debug extensions allow the core to be stopped either on a given instruction fetch (breakpoint) or data access (watchpoint). When the core is stopped by connecting an external debugger to a serial wire debug (SWD) interface, the internal core state and the external system state may be examined. Once examination is completed, the core and the system may be restored, and program execution may be resumed.

The GR5526 supports the following debug and trace features:

- Two-pin SWD interface
- Flash patch and breakpoint (FPB) unit support
  - Two literal comparators
  - Sixteen instruction comparators
- Data watchpoint trigger (DWT) unit
- Instrumentation trace and macrocell (ITM)
- Trace port interface unit (TPIU)
  - 4-bit parallel trace of ITM trace data
  - Serial wire output (SWO) trace of ITM data

3.6 Memory Subsystem

The GR5526 integrates multiple kinds of memory as shown below:

- 512 KB SRAM
- 1 MB Flash
- 640 KB ROM
- 512-byte eFuse
- 8 KB Cache
- 8 MB PSRAM (internal or external)

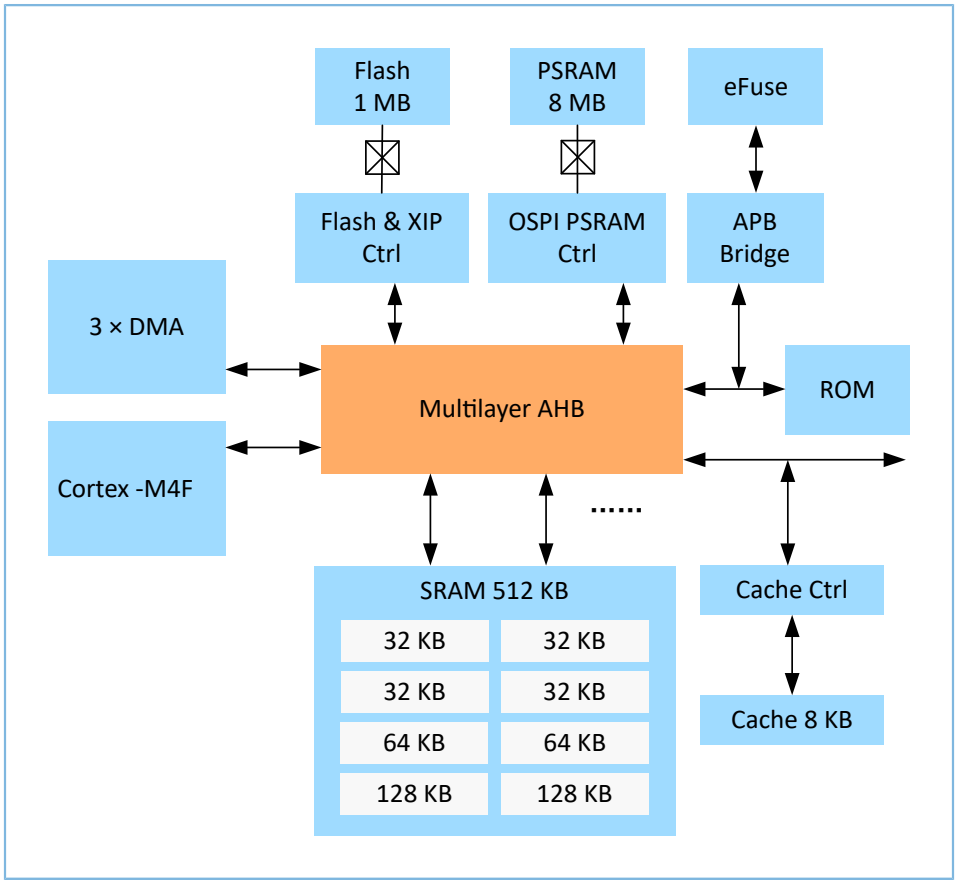

Figure 3-3 Memory subsystem

3.6.1 SRAM

The main Static Random Access Memory (SRAM) is comprised of an on-chip static RAM totaling up to 512 KB. The main SRAM is implemented as several SRAM instances to allow for more control of power usage when less SRAM is required (4 x 32 KB instances, 2 x 64 KB instances, 2 x 128 KB instances). Each SRAM instance has one or two independent power switches, which control SRAM in three states: full power, retention power, and power off. The relationship between SRAM instances and power domains is shown in Table 4-4.

3.6.1.1 Introduction

Although always contiguous on all GR5526 devices, the SRAM instances are divided between eight AHB matrix ports. This allows user programs to potentially achieve better performance by dividing RAM usage among the ports. For example, simultaneous access to SRAM0 by the CPU and SRAM1 by the system DMA controller does not result in any bus stalls for either master. When multiple masters access the same SRAM, GR5526 uses round-robin scheduling to configure the priorities of multiple hosts, ensuring the fairness of multi-master access.

Generally, the CPU will read or write all peripheral data at some point, even when all such data is read from or sent to a peripheral by DMA. Therefore, minimizing stalls is likely to involve putting data to/from different peripherals in RAM on each port.

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

Alternatively, sequences of data from the same peripheral can be alternated between RAM on each port. This can be helpful if DMA fills or empties a RAM buffer, and then signals the CPU before proceeding with a second buffer. The CPU will then tend to access the data while the DMA is using RAM on the other port.

At the same time, according to Section 3.3 Memory Map, GR5526 provides an Aliasing Memory with the start address of **0x0010_0000** and **0x3000 0000** for the RAM with the start address of **0x2000 0000**.

- The start address of the area is **0x2000 0000**, which supports bit-segment operations, and its corresponding bit-segment start address is **0x2200 0000**.
- Due to the characteristics of the Cortex®-M4F bus architecture, the access efficiency of the area with the start address of **0x0010 0000** is higher than that of other areas, so the running address of the RAM_CODE executable code is within the address range of the **0x0010_0000** area, which can speed up the operation. RW, ZI, HEAP, and STACK use RAM in the **0x2000 0000** area.

3.6.2 eFuse

3.6.2.1 Introduction

GR5526 eFuse (512 bytes) stores the product ID, TRIM, and security parameters. It consists of Main Region (240 bytes), Backup Region (240 bytes), and User Region (32 bytes). User Region is reserved for users to write and read data by calling the eFuse driver. Every single byte of eFuse can only be written once. Figure 3-4 shows the structure of eFuse.

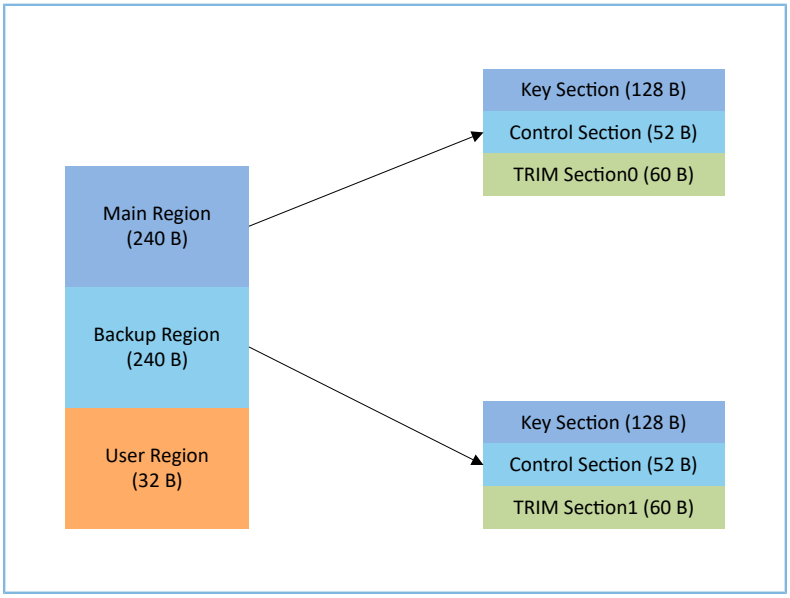

Figure 3-4 eFuse structure

Note:

The access to eFuse should be word-aligned, which means the offset address can only be 0x0, 0x4, …, 0x1C.

3.6.2.2 Main Features

- The initial eFuse bit is zero, and it can only be changed once from zero to one.
- Little-endian mode; the minimum unit size of read or write is 4 bytes.
- The Main Region has three sections: Key section, Control section and TRIM section0.
  - Key section stores all the keys needed for Secure Boot and SPI Flash encryption.
  - Control section stores some parameters required by the boot loader for chip configuration.
  - TRIM section0 stores chip UID, TRIM parameters, and product configurations. Users can read the chip UID from the address **0x40016110**.
- The Backup Region has three sections: Key section, Control section and TRIM section1.
  - The values in Control section and Key section of the Backup Region are the same as those in the Main Region.
  - TRIM section1 is used to save Flash parameters as well as ADC and COMP calibration parameters.
- The User Region is reserved for users to write and read data by calling the eFuse driver.
  - The base address of User Region is **0x40016000**.
  - Every single byte of eFuse can only be written once.
  - Reading from or writing to User Region requires disabling main_or_backup.

Users can use GProgrammer to create and download eFuse files.

3.6.2.3 eFuse Information

Table 3-5 eFuse information

Name	Offset	Size (Byte)	Description
Product ID	0x15E	0x2	Product ID; the default value is 0x0000; customers can write this field.
Chip ID	0x158	0x6	Chip ID; the default value is 0x000000000000; customers can write this filed.
EncMode	0x152	0x2	Firmware security mode (encryption and signature verification) • 0000 0000b: Firmware normal mode • Others: Firmware security mode
SWDDisable	0x150	0x2	Bit0–Bit15: Disable SWD. • 0000 0000b: SWD enabled • Others: SWD disabled
Config	0x14C	0x4	• Bit0: In-system programming (ISP) process by UART ◦ 0b: UART ISP process is supported. ◦ 1b: UART ISP process is not supported. • Bit1: ISP process by USB ◦ 0b: USB ISP process is supported.

Name	Offset	Size (Byte)	Description
			◦ 1b: USB ISP Process is not supported. • Bit2: ISP process by J-Link ◦ 0b: J-Link ISP process is supported. ◦ 1b: J-Link ISP process is not supported. • Bit3–Bit5: In security mode, system clock frequency during cold boot ◦ 000b: XO 16 MHz ◦ 001b: PLL 64 MHz ◦ 010b: PLL 96 MHz ◦ 011b: PLL 48 MHz ◦ 100b: PLL 24 MHz ◦ 101b: PLL 16 MHz • Bit6–Bit8: In security mode, XQSPI clock frequency during cold boot ◦ 000b: 16 MHz ◦ 001b: 48 MHz ◦ 010b: 32 MHz ◦ 011b: 24 MHz ◦ 100b: 16 MHz ◦ 101b: 64 MHz • Bit9–Bit16: XQSPI XIP read mode ◦ 0000 0011b: Read mode ◦ 0000 1011b: Fast read mode ◦ 0011 1011b: DualOut fast read mode ◦ 0110 1011b: QuadOut fast read mode ◦ 1011 1011b: DualIO fast read mode ◦ 1110 1011b: QuadIO fast read mode • Bit17: Memory size ◦ 0b: 512 KB ◦ 1b: 288 KB • Bit18–Bit31: Reserved
Package Type	0x134	0x2	• Bit0–Bit4: Chip type ◦ - 0x0100: GR5526VGBIP - 0x0101: GR5526VGBI - 0x0200: GR5526RGNIP

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

Name	Offset	Size (Byte)	Description
			- 0x0201: GR5526RGNI ◦ Others: Reserved
XO	0x12A	0x2	Bit0-Bit15: XO offset
BT_MAC	0x124	0x6	Bluetooth device address; the system will find the address in NVDS (Non-Volatile Data Storage) first; if no address is found in NVDS, then use this address as the device address.
IO_LDO_SEL	0x123	0x1	• Bit0: IO_LDO selector ◦ 0b: LDO voltage is 1.8 V. ◦ 1b: LDO voltage is 3.3 V. • Bit1: IO_LDO bypass ◦ 0b: bypass disabled ◦ 1b: bypass enabled
Chip UUID	0x110	0x10	• Byte0–Byte2: FAB • Byte3: Year • Byte4: Month • Byte5–Byte12: LOT_ID • Byte13: Wafer_ID • Byte14: Wafer_X • Byte15: Wafer_Y
User region	0x000	0x20	eFuse user region; customers can write this field.

3.6.2.4 Registers

3.6.2.4.1 EFUSE_TPGM

- **Name:** EFUSE TPGM Register
- **Description:** EFUSE TPGM Register
- **Base Address:** 0x4001_6400
- **Offset:** 0x00
- **Reset Value:** 0x100000F0

Table 3-6 EFUSE_TPGM Register

Bits	Field Name	RW	Reset	Description
31:24	EFUSE_WR_INR	RW	0x10	Interval between two bits in eFuse, number of APB CLK cycles

Bits	Field Name	RW	Reset	Description
11:0	EFUSE_PGM_T	RW	0xF0	eFuse program time, tpro = EFUSE_PGM_T*T, where T is clock period; range: 9000 ns < tpro < 11000 ns (typical: 10000 ns)

3.6.2.4.2 EFUSE_PGENB

- **Name:** EFUSE PGENB Register
- **Description:** EFUSE PGENB Register
- **Base Address:** 0x4001_6400
- **Offset:** 0x04
- **Reset Value:** 0x00000001

Table 3-7 EFUSE PGENB Register

Bits	Field Name	RW	Reset	Description
0	PGENB	RW	0x1	eFuse PGENB signal

3.6.2.4.3 EFUSE_OPTION

- **Name:** EFUSE OPTION Register
- **Description:** EFUSE OPTION Register
- **Base Address:** 0x4001_6400
- **Offset:** 0x08
- **Reset Value:** 0x00000000

Table 3-8 EFUSE OPTION Register

Bits	Field Name	RW	Reset	Description
0	EFUSE_INIT_CHK_START	WO	0x0	Writing "1" to this bit enables eFuse ctrl to start to read the whole eFuse value; writing “0” to this bit enables checking this value.

3.6.2.4.4 EFUSE_STAT

- **Name:** EFUSE STAT Register
- **Description:** EFUSE STAT Register
- **Base Address:** 0x4001_8400
- **Offset:** 0x10

- **Reset value:** 0x00000000

Table 3-9 EFUSE STAT Register

Bits	Field Name	RW	Reset	Description
2	EFUSE_WR_DONE	RC	0x0	1: eFuse one word write done **Note:** Reading this register enables hardware to clear this bit.
1	EFUSE_INIT_CHK_OK	RC	0x0	1: eFuse initial value check success which means the eFuse values are all "0" **Note:** Reading this register enables hardware to clear this bit.
0	EFUSE_INIT_CHK_DONE	RC	0x0	1: eFuse initial value check done **Note:** Reading this register enables hardware to clear this bit.

3.6.2.4.5 RF_REG_2

- **Name:** RF_REG_2 Register
- **Description:** RF_REG_2 Register
- **Base Address:** 0x4000A800
- **Offset:** 0x0C
- **Reset Value:** 0x0000A0C0

Table 3-10 RF_REG_2 Register

Bits	Field Name	RW	Reset	Description
12	EFUSE_VDDQ_EN_DLY	RW	0x0	Delayed enable signal for efuse_vddq (2.5 V)
11	EFUSE_VDDQ_EN	RW	0x0	Enable eFuse 2.5 V.
10	EFUSE_VDD_EN	RW	0x0	Enable eFuse 1.1 V.

3.6.3 Flash (Non-Volatile Memory)

The non-volatile memory management system integrates an accessible Flash memory, on-chip QSPI controller, and XIP controller – all allow GR5526 SoCs to be capable of storing application data and code execution in Flash address.

The XIP mechanism allows the CPU to execute program code directly from the internal Flash memory by using the QSPI interface. When a cache miss occurs, the XIP controller will automatically prefetch a certain number of program instructions from Flash memory and update them into the cache memory.

3.6.3.1 Write and Erase

Flash memory can be read by the CPU unlimited times through the XIP controller, but the number of times it can be written or erased (up to 100000 cycles) and how it can be written are restricted.

Writing to Flash memory is managed by the on-chip QSPI controller. Before performing a write/erase, the XIP controller must be disabled for writing and the on-chip QSPI controller must be enabled. After writing/erasing, software needs to restore the XIP controller. Users must make sure that writing and erasing are not performed at the same time. If the CPU executes code from Flash while the on-chip QSPI controller is writing to Flash, the CPU execution will be stalled. In this case, the SRAM should be used for code execution.

The Flash memory is only able to write '0' to erased bits, but it cannot write a bit back to '1'. Therefore, a page action must be performed before a write to the same address in Flash. GR5526 supports write operations both in encryption mode and non-encryption mode. The data written in encryption mode is protected from being read arbitrarily. In encryption mode, Flash is written by a full 32-bit word to a word-aligned address in Flash. At the same time, write and erase operations of the Flash are 4 KB as a loop to respond to external interrupts, so as to avoid operations occupying CPU for a long time and causing the external process being crowded.

After a Flash page is erased, all bits in the page are set to '1'. The time required for erasing is determined by the electrical parameters of the Flash, as shown in Table 3-11. The CPU is stalled if the CPU executes code from the Flash while the controller performs erase operations.

3.6.3.2 Flash Operation Interfaces

All the Flash operation interfaces have been provided by the GR5526 SDK package to ensure security of Flash operations. Users do not need to care about the implementation details of Flash operations.

3.6.3.3 Electrical Specifications

Table 3-11 Electrical specifications

Description	Min.	Typ.	Max.	Unit
Erase Suspend Latency			30	µs
Program Suspend Latency			30	µs
Latency between Program Resume and next Suspend	0.3			µs
Latency between Erase Resume and next Suspend	0.3			µs
Page Program time (up to 256 bytes)		8	20	ms
Sector Erase time		8	20	ms
Chip Erase time		8	20	ms
Program/Erase Cycles		100000		cycles

Note:

If the data length (L_{data_length}, 4-byte aligned) in each Flash Program is longer or equal to 40 bytes, the Flash max Erase cycle is 100000; if the Program data length is less than 40 bytes, you can calculate the max Erase cycle base on the formula:

$$N_{max_erase_cycle} = 10000 \times \left(\frac{L_{data_length}}{4}\right) \quad (1)$$

For example, if the data length in each Flash Program is 4, then the max Erase cycle is 10000. If your product Flash Erase cycle is higher than the result from the formula, you can add data cache to reduce the Flash Program times or trade more Flash space for Flash Program times.

3.6.4 Cache

The GR5526 cache supports:

- 4-way set associative map mode with LFU replacement policy
- 8 KB cache memory (DataRAM) and 16 bytes per cache line
- Auto clock gating for power saving
- Profiling counters

GR5526 has an 8 KB cache memory that supports two map modes: direct map and 4-way set associative map with least frequently used (LFU) replacement policy. Both instruction and data access towards Flash memory or XIP code regions can be cached. The cache memory is enabled by default, but can be disabled by setting CACHE_DIS bit in XF_CTRL register. When cache is disabled, the number of Flash reading increases significantly, thus increasing power consumption and reducing the access speed.

The cache consists of TagRAM and DataRAM. The TagRAM is used to match addresses, and DataRAM is used to cache data. One cache line has 128 bits (16 bytes) DataRAM to cache data. In direct map mode, there are 512 cache lines; in 4-way set associative mode, one way has 128 cache lines.

The cache includes profiling counters. To measure the performance (hit-rate) of a code-section, the built-in profiling counters should be enabled. Before the beginning of the section, start the profiling counters by resetting CLR_HITMISS bit in XF_CTRL register. At the end of the section, the number of cache hits and cache misses for the section can be read from XF_HIT_COUNT and XF_MISS_COUNT registers, and users can stop the counters by setting CLR_HITMISS bit. The cache hit-rate can be calculated based on the formula: XF_HIT_COUNT/(XF_HIT_COUNT + XF_MISS_COUNT). The XF_HIT_COUNT and XF_MISS_COUNT registers do not wrap around after reaching the maximum value. If the maximum value is reached, consider profiling for a shorter duration to get correct numbers.

3.6.4.1 Registers

3.6.4.1.1 XF_CTRL

- **Name**: Cache Control Register
- **Description**: Cache Control Register

- **Base address**: 0x4000D000
- **Offset**: 0x00
- **Reset value**: 0x00000012

Table 3-12 Cache Control Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R		Reserved bits
11	DIREC_MAP	RW	0x0	Enable direct map. • 0x0: 4-way associative map enabled • 0x1: Direct map enabled (no effect for sequential access)
10	CLK_FORCE_EN_3	RW	0x0	Cache_FSM and cache memory clock gating • 0x0: Forcibly disable gating. • 0x1: Enable gating dynamically.
9	CLK_FORCE_EN_2	RW	0x0	APB clock gating control • 0x0: Forcibly disable gating. • 0x1: Enable gating dynamically.
8	CLK_FORCE_EN_1	RW	0x0	Cache CLK_96 gating control • 0x0: Forcibly disable gating. • 0x1: Enable gating dynamically.
7	CLK_FORCE_EN_0	RW	0x0	XQSPI and cache HCLK gating control • 0x0: Forcibly disable gating. • 0x1: Enable gating dynamically.
6	BUF_DIS	RW	0x0	Disable cache buffer preload. • 0x0: Cache preload enabled • 0x1: Cache preload disabled
5	DIS_SEQ_ACCESS	RW	0x0	Disable sequential cache access. • 0x0: Cache sequential access enabled • 0x1: Cache sequential access disabled
4	CLR_HITMISS	RW	0x1	Clear hit/miss counters. • 0x0: Hit and miss counters in normal mode • 0x1: Hit and miss counters in clear mode (default)
3	CLR_FIFO	RW	0x0	Clear LFU FIFO. • 0x0: FIFO in normal mode (default) • 0x1: FIFO in clear mode
2	RSVD	R	0x0	Reserved bit

Bits	Field Name	RW	Reset	Description
1	TAG_MEM_FLUSH_EN	RW	0x1	Enable tag memory flush. Out of resetting tag memory will get flushed unless tag_ret is enabled (tag_ret signal lives in CPU register space). • 0x0: Tag memory flush is disabled. • 0x1: Tag memory flush is enabled.
0	CACHE_DIS	RW	0x1	Disable cache. • 0x0: Cache enabled • 0x1: Cache disabled (default)

3.6.4.1.2 XF_HIT_COUNT

- **Name**: Cache Hit Count Register
- **Description**: Cache hit counter
- **Base Address**: 0x4000D000
- **Offset**: 0x08
- **Reset Value**: 0x00000000

Table 3-13 Cache Hit Count Register

Bits	Field Name	RW	Reset	Description
31:0	CACHE_HIT_CNT	RO	0x0	Read cache hit count (If bit[4] of XF_CTRL is 0).

3.6.4.1.3 XF_MISS_COUNT

- **Name**: Cache Miss Count Register
- **Description**: Cache miss counter
- **Base Address**: 0x4000D000
- **Offset**: 0x0C
- **Reset Value**: 0x00000000

Table 3-14 Cache Miss Count Register

Bits	Field Name	RW	Reset	Description
31:0	CACHE_MISS_CNT	RO	0x0	Read cache miss count (If bit[4] of XF_CTRL is 0).

3.6.4.1.4 XF_STAT

- **Name**: Cache Flush Status Register

- **Description**: Read cache flush status.
- **Base Address**: 0x4000D000
- **Offset**: 0x10
- **Reset Value**: 0x00000000

Table 3-15 Cache Flush Status Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	RO	0x0	Reserved bits
0	CACHE_STAT	RO	0x0	0x1: Tag flush busy 0x0: Tag flush not busy

3.6.5 PSRAM

The GR5526VGBIP and GR5526RGNIP SoCs have an 8 MB PSRAM. PSRAM uses OSPI for data access. The OSPI address mapping is used to map PSRAM space to **0x30080000** area, which is helpful for users to access this memory space. Pin definitions are shown in Table 3-16.

Table 3-16 Pin definitions

Symbol	Type	Description
VDD	Power	Core supply 1.8 V
VDDQ	Power	IO supply 1.8 V
VSS	Ground	Core supply ground
VSSQ	Ground	IO supply ground
A/DQ[7:0]	IO	Address/DQ bus[7:0]
DQS/DM	IO	DQ Strobe (DQS) clock during reads, data mask during writes. DM (Data Mask) is active high. DM = 1 means “do not write”.
CE#	Input	Chip select, active low. When CE# = 1, chip is in standby state.
CLK	Input	Clock signal

3.6.5.1 Specifications

- Single supply voltage:
  - VDD: 1.62 to 1.98 V
  - VDDQ: 1.62 to 1.98 V
- Interface: OSPI, two bytes transferred per one clock cycle
- Performance: Only 48 MHz clock rate is supported, up to 70 MB/s read/write throughput
- Organization: 8 MB with 1024 Bytes page size

- Refresh: Self-managed
- Operating temperature range:
  - Tc: -40°C to +85°C (standard range)
- Maximum standby current:
  - 300 µA @ 105°C
  - 200 µA @ 85°C
  - 100 µA @ 25°C
  - 30 µA @ 25°C (half sleep mode with data retained)

3.6.5.2 Features

- Low power features
  - Partial array self-refresh
  - Auto Temperature Compensated Self-Refresh (ATCSR) by built-in temperature sensor
  - User-configurable refresh rate
  - Ultra Low Power (ULP) in half sleep mode with data retained
- Software reset
- Output driver LVCMOS with programmable drive strength
- Data Mask for writing data
- Data Strobe enables high-speed read operation
- Register-configurable write and read initial latencies
- Write burst length with a maximum of 1024 bytes and a minimum of 2 bytes
- Wrap and hybrid burst in 16 B/32 B/64 B/1 KB lengths
- Linear burst command
- Row boundary crossing (RBX)
  - Read operations can be enabled via Mode Register
  - RBX write is not supported

3.6.5.3 Functional Description

PSRAM is byte-addressable. Memory accesses are required to start on both even and odd addresses, but starting memory access at even addresses is more efficient than starting memory access at odd addresses (example, A[0] = 0). Mode Register accesses allow both even and odd addresses.

The modes of read and write operations are in default Hybrid Wrap 32 mode. Other burst lengths of 16 B, 32 B, 64 B or 1 KB in standard or hybrid wrap modes are configurable by using registers, as shown in the table below. The PSRAM

also includes command for linear bursting. Bursts can start on any even address. Write burst length has a minimum of 2 bytes. Read has no minimum length.

Table 3-17 Wrap mode configuration

Register MR8[2]	Register MR8[1:0]	Burst Length	Example of Sequence of Bytes During Wrap	
			Starting Address	Byte Sequence
0b'0	0b'00	16 Byte Wrap	4	[4,5,6,...,15,0,1,2,3,...]
0b'0	0b'01	32 Byte Wrap	4	[4,5,6,...,31,0,1,2,3,...]
0b'0	0b'10	64 Byte Wrap	4	[4,5,6,...,63,0,1,2,3,...]
0b'0	0b'11	1K Byte Wrap	4	[4,5,6,...,1023,0,1,2,3,...]
0b'1	0b'00	16 Byte Hybrid Wrap	2	[2,3,4,...,15,0,1],16,17,...,1023, 0,1,...]
0b'1	0b'01	32 Byte Hybrid Wrap	2	[2,3,4,...,31,0,1],32,33,...,1023, 0,1,...]
0b'1	0b'10	64 Byte Hybrid Wrap	2	[2,3,4,...,63,0,1],64,65,...,1023, 0,1,...]
0b'1	0b'11	1k Byte Hybrid Wrap	2	[2,3,4,...,1023,0,1,2,...]

PSRAM supports the following instructions and follows a certain timing.

Table 3-18 Supported instructions

	1st CLK	2nd CLK		3rd CLK	
	Rising/Falling edge	Rising edge	Falling edge	Rising edge	Falling edge
Command	Instruction	Address[31:24]	Address[23:16]	Address[15:8]	Address[7:0]
Sync Read	00h	A3	A2	A1	A0
Sync Write	80h	A3	A2	A1	A0
Sync Read (Linear Burst)	20h	A3	A2	A1	A0
Sync Write (Linear Burst)	A0h	A3	A2	A1	A0
Mode Register Read	40h	-	-	-	Mode Register Address
Mode Register Write	C0h	-	-	-	Mode Register Address
Global Reset	FFh	-	-	-	-

The PASR bits restrict refresh operations to a portion of the total memory array. This feature allows the device to reduce standby current by refreshing only that part of the memory array required by the host system. The refresh options are full array, one-half array, one-quarter array, one-eighth array, or none of the array. The mapping of these partitions can start at either the beginning or the end of the address map.

Table 3-19 Self refresh zone configuration

Register MR4[2:0]	Refresh Coverage	Address Space	Size
0b’000	Full array (default)	000000h-7FFFFFh	8 MB
0b’001	Bottom 1/2 array	000000h-3FFFFFh	4 MB
0b’010	Bottom 1/4 array	000000h-1FFFFFh	2 MB
0b’011	Bottom 1/8 array	000000h-0FFFFFh	1 MB
0b’100	None	0	0 MB
0b’101	Top 1/2 array	400000h-7FFFFFh	4 MB
0b’110	Top 1/4 array	600000h-7FFFFFh	2 MB
0b’111	Top 1/8 array	700000h-7FFFFFh	1 MB

Half Sleep mode supports putting the device in an ultra-low power state, while the stored data is retained. Half Sleep Mode Entry is entered by writing **0xF0** into Register MR6. CE# going high initiates the Half Sleep mode and must be maintained for the minimum duration of t_{HS} (minimum Half Sleep duration). Half Sleep Exit is initiated by a low-pulsed CE#.

Deep Power Down (DPD) mode supports putting the device into power down state. DPD Entry is entered by writing 0xC0 into Register MR6. CE# going high initiates the DPD mode and must be maintained for the minimum duration of t_{DPD} (minimum Deep Power Down duration). DPD Exit is initiated by a low-pulsed CE#. Register values and memory content are not retained in DPD mode. Afterwards, the DPD mode register values will be reset to defaults. t_{DPDp} is minimum period between two DPD modes (measured from DPD Exit to the next DPD Entry) as well as from the initial power-up to the first DPD Entry.

3.6.5.4 Registers

3.6.5.4.1 MR0

- **Name**: Mode Register 0
- **Description**: Mode Register
- **Base Address**: OSPI_XIP_BASE_ADDRESS
- **Offset**: 0x00
- **Reset value**: 0x09

Table 3-20 MR0 Register

Bits	Field Name	RW	Reset	Description
7:6	FX	RW	0x0	Fixed
5	RD_LT	RW	0x0	Read latency type **Value**: • 0x0: Variable (default)

Bits	Field Name	RW	Reset	Description
				• 0x1: Fixed
4:2	RD_LC	RW	0x2	Read latency code When MR0[5] = 0, the latency value and max push out value for each bit of MR0[4:2] are listed below. • 0x0: 3, 6 • 0x1: 4, 8 • 0x2: 5 (default), 10 • 0x3: 6, 12 • 0x4: 7, 14 • Others: None When MR0[5] = 1, the latency value for each bit of MR0[4:2] is listed below. • 0x0: 6 • 0x1: 8 • 0x2: 10 • 0x3: 12 • 0x4: 14 • Others: None The standard/extended Max. Input CLK Freq (MHz) value for each bit of MR0[4:2] is listed below. • 0x0: 66 • 0x1: 109 • 0x2: 133 • 0x3: 166 • 0x4: 200 • Others: None
1:0	DS	RW	0x1	Drive Strength **Value**: • 0x0: Full (25 Ω) • 0x1: Half (50 Ω, default) • 0x2: 1/4 (100 Ω) • 0x3: 1/8 (200 Ω)

3.6.5.4.2 MR1

- **Name**: Mode Register 1
- **Description**: Mode Register
- **Base Address**: OSPI_XIP_BASE_ADDRESS
- **Offset**: 0x01
- **Reset Value**: 0x8D

Table 3-21 MR1 Register

Bits	Field Name	RW	Reset	Description
7	ULP	R	0x0	Ultra Low Power Device **Value**: • 0x0: Non-ULP (no Half Sleep) • 0x1: ULP (Half Sleep supported)
6:5	RSVD	R		Reserved bits
4:0	VID	R	0x2	Vendor ID • 0xD: APM

3.6.5.4.3 MR2

- **Name**: Mode Register 2
- **Description**: Mode Register
- **Base Address**: OSPI_XIP_BASE_ADDRESS
- **Offset**: 0x02
- **Reset Value**: 0x93

Table 3-22 MR2 Register

Bits	Field Name	RW	Reset	Description
7	GB	R	0x1	Good-Die **Value**: • 0x0: Fail • 0x1: Pass
6:5	RSVD	R		Reserved bits
4:3	DID	R	0x2	Device ID • 0x0: Generation 1 • 0x1: Generation 2

Bits	Field Name	RW	Reset	Description
				• 0x2: Generation 3 (default) • Others: Reserved
2:0	DEV_D	R	0x3	Device Density • 0x1: 4 MB • 0x3: 8 MB (default) • 0x5: 16 MB • 0x7: 32 MB • Others: Reserved

3.6.5.4.4 MR3

- **Name**: Mode Register 3
- **Description**: Mode Register
- **Base Address**: OSPI_XIP_BASE_ADDRESS
- **Offset**: 0x03
- **Reset Value**: 0x80

Table 3-23 MR3 Register

Bits	Field Name	RW	Reset	Description
7	RBX	R	0x1	Row boundary crossing enable **Value**: • 0x0: RBX not supported • 0x1: RBX supported via MR8[3] = 1
6	VCC	R	0x0	Operating voltage range **Value**: • 0x0: 1.8 V (default) • 0x1: 3 V
5	SRF	R	0x0	Self refresh flag **Value**: • 0x0: Slow refresh (allowed via MR4[3] = 1, otherwise fast refresh) • 0x1: Fast refresh
4:0	RSVD	R		Reserved bits

3.6.5.4.5 MR4

- **Name**: Mode Register 4
- **Description**: Mode Register
- **Base Address**: OSPI_XIP_BASE_ADDRESS
- **Offset**: 0x04
- **Reset Value**: 0x40

Table 3-24 MR4 Register

Bits	Field Name	RW	Reset	Description
7:5	WR_L	RW	0x2	Write latency The values of write latency and Fmax (MHz) for MR4[7:5] are listed below. • 0x0: 3, 66 • 0x4: 4, 104 • 0x2: 5 (default), 133 • 0x6: 6, 166 • 0x1: 7, 200 • Others: reserved, none
4	FX	RW	0x0	Fixed
3	FLUSH_FREQ	RW	0x0	Refresh Frequency **Value**: • 0x0: Fast refresh (default) • 0x1: Enable slow refresh when temperature allows
2:0	PASR	RW		Partial array self-refresh **Value**: Refer to Table 3-19

3.6.5.4.6 MR6

- **Name**: Mode Register 6
- **Description**: Mode Register
- **Base Address**: OSPI_XIP_BASE_ADDRESS
- **Offset**: 0x06
- **Reset Value**: 0x00

Table 3-25 MR6 Register

Bits	Field Name	RW	Reset	Description
7:4	ULP_MODE_SEL	W	0x0	ULP modes selection

Bits	Field Name	RW	Reset	Description
				• 0xF: Half Sleep mode • 0xC: Deep Power Down mode • Others: Reserved
3:0	RSVD	W		Reserved bits

3.6.5.4.7 MR8

- **Name**: Mode Register 8
- **Description**: Mode Register
- **Base Address**: OSPI_XIP_BASE_ADDRESS
- **Offset**: 0x08
- **Reset Value**: 0x00

Table 3-26 MR8 Register

Bits	Field Name	RW	Reset	Description
7	FX	RW	0x0	Fixed
6:4	RSVD	R		Reserved bits
3	RBX	RW	0x0	Row Boundary Crossing Read Enable **Value**: • 0x0: Reads stay within the 1K column address space • 0x1: Reads cross row at 1K boundaries
2	BT	RW	0x1	Burst Type. **Value**: Refer to Table 3-17
1:0	BL	RW	0x1	Burst Length **Value**: Refer to Table 3-17

4 PMU

4.1 Functional Overview

The Power Management Unit (PMU) is responsible for generating all required voltages for different blocks in GR5526. It is optimized for application scenarios with ultra-low power requirements in order to ensure the maximum power efficiency.

In active mode, a DC-DC converter generates voltage for both the transceiver and the DIG_CORE_LDO (LDO) which generates voltage for digital blocks. GR5526 also supports using the DC-DC converter to directly power the digital blocks instead of the DIG_CORE_LDO (LDO). However, since voltage of the transceiver is different from that of the digital blocks, developers need to provide appropriate voltages to different modules. Usually the transceiver requires higher voltage than the digital blocks, so voltage of the DC-DC converter needs to be temporarily increased to meet the voltage requirement of the transceiver when the transceiver is working. The voltage adaptation of the DC-DC converter has already been implemented in the integrated GR5526 SDK.

In order to reduce the power consumption of the chip during sleep, GR5526 uses an LP_LDO to supply its AON modules that stay ON when both the MCU subsystem and the Bluetooth LE subsystem are OFF. The LP_LDO also generates a lower voltage for content retention to the memories where their content is needed after wakeup.

Both the digital voltage and the retention voltage are connected to all power islands through a control switch matrix on the chip.

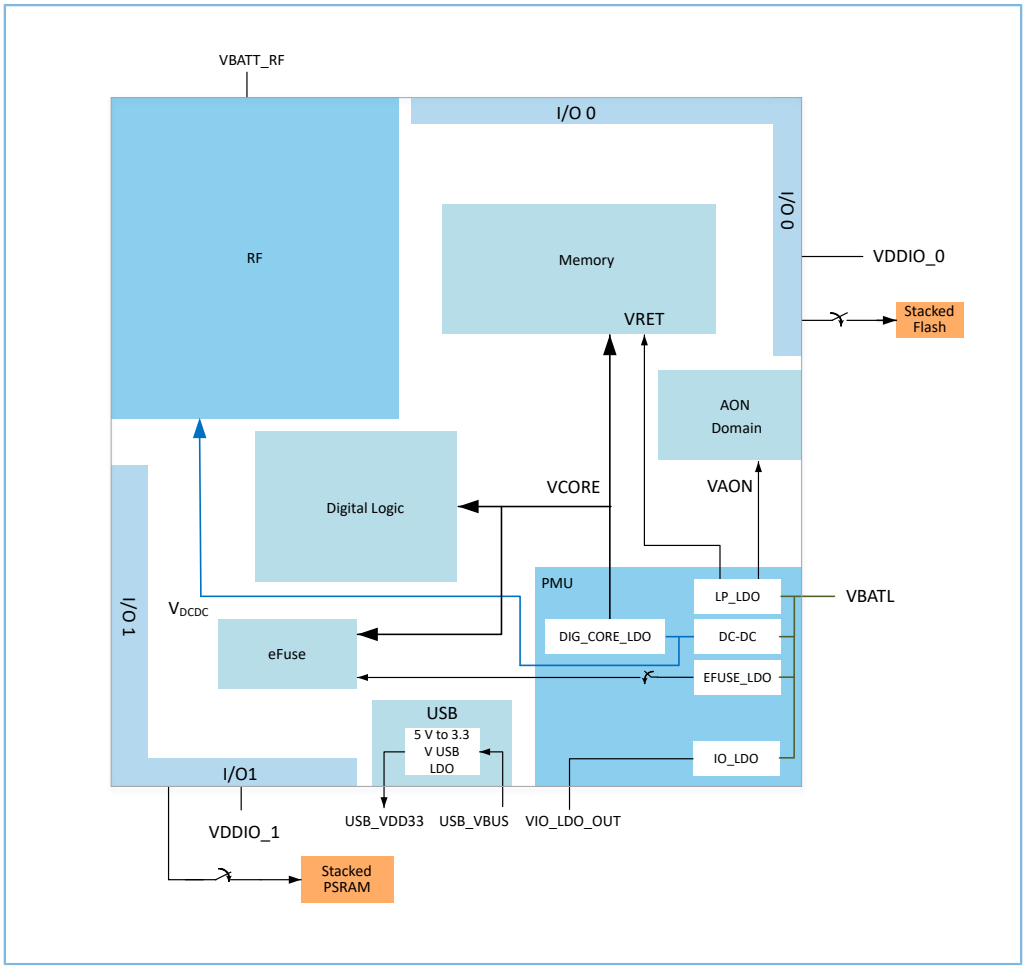

Figure 4-1 Power management block diagram

As shown in Figure 4-1, the GR5526 power architecture is based on the PMU that generates the following supplies from an external supply VBATL.

- IO_LDO: Used to generate the IO voltage to supply the pads of GR5526 and the external devices connecting to GR5526. The IO_LDO is also used to supply power to the stacked Flash.
- VDDIO0: Supply for I/O0 group, connected to stacked Flash and set to voltage required by Flash. This pin requires voltage irrigation, which can be powered by the chip's internal power supply or by an external power supply.
- VDDIO1: Supply for I/O1 group, connected to stacked PSRAM and set to voltage required by PSRAM. This pin requires voltage irrigation, which can be powered by the chip's internal power supply or by an external power supply.
- LP LDO: Supply for AON block, and retention supply for memory instances
- DIG_CORE_LDO: Supply for digital logic blocks
- DC-DC: Supply for RF domain and digital core IDO. Supply for digital logic blocks conditionally.
- EFUSE_LDO: Supply for eFuse programming. eFuse usually needs a high voltage about 2.5 V during programming. DIG_CORE_LDO only controls the digital logic of eFuse.

- VBAT_RF: Usually connected to VBATL
- The USB analog module is powered by a 5 V external power supply through USB_VBUS. The embedded USB LDO converts 5 V to 3.3 V, which is needed for level translation of internal USB differential pair signaling.

4.1.1 Power Management

To minimize power consumption, the GR5526 SoC supports power management in multiple power modes, including active mode, IDLE mode, deep sleep mode, and ultra deep sleep mode.

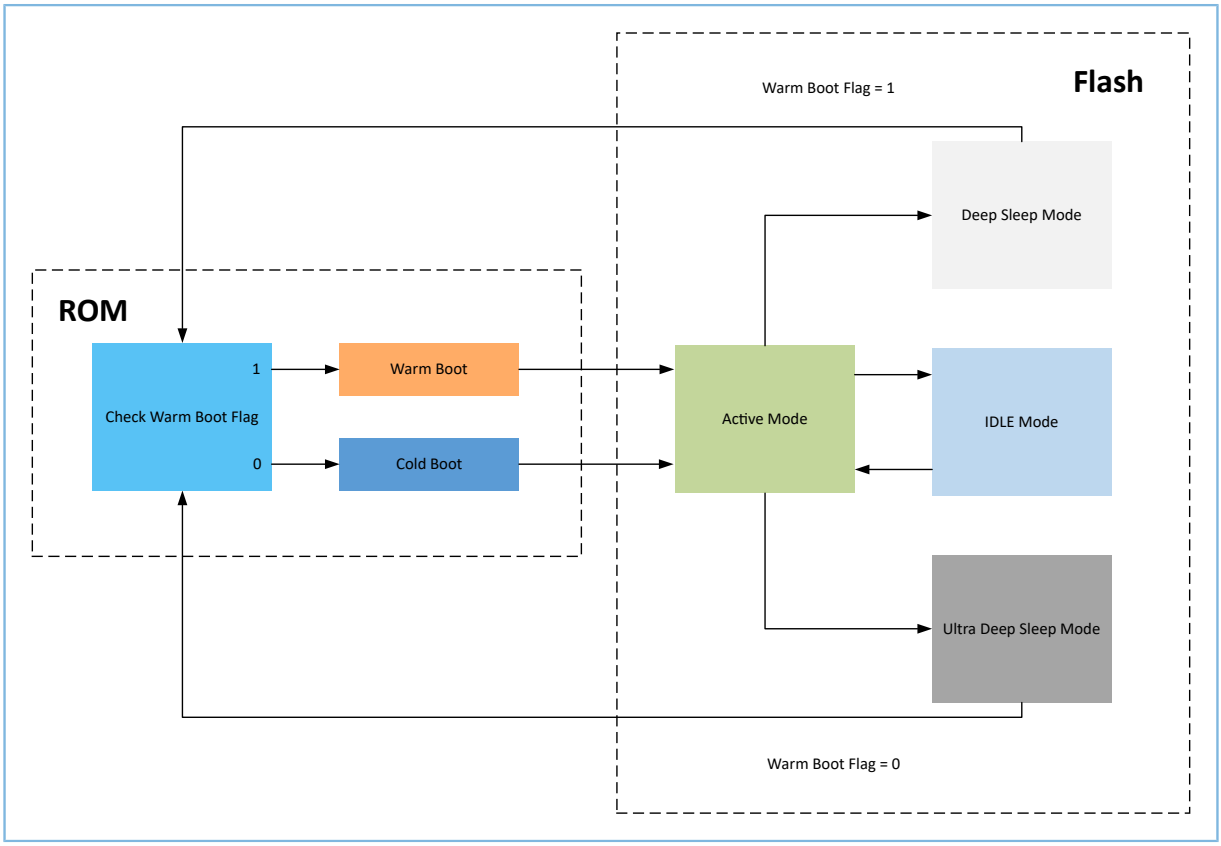

Figure 4-2 Power mode block diagram

- Cold Boot: Start the MCU from power-off or reset status. Current configuration and data are discarded. Program processing begins again with the initial values.
- Warm Boot: Restart the MCU from sleeping. Program processing starts once again where retentive configuration and data are retained.
- Check Warm Boot Flag: Whether to enter the warm boot can be judged by the warm boot flag in the SOFTWARE_REG0.Bit[8] register (Address: 0xA000A340).

The power management in different power modes for different modules is shown in Table 4-1.

Table 4-1 GR5526 power mode management

Module	Software-Configurable Power Modes				Reset Pin Held
	Active	IDLE	Deep Sleep	Ultra Deep Sleep	
CPU	Active	Off	Off	Off	Off
Flash	On	Standby	Off	Off	Off
SRAM	On	On	Retention	Off	Off
Radio	Available	Available	Off	Off	Off
Supply System	On	On			Off
Current			3.3 μA	1.3 μA	0.15 μA
Register Retention	Full	Full	No	No	No
High-Speed Clock	HFXO_32M or HFRC_192M	HFXO_32M or HFRC_192M	Off	Off	Off
Low-Speed Clock	LFXO_32K or LFRC_32K	LFXO_32K or LFRC_32K	LFXO_32K or LFRC_32K	LFXO_32K or LFRC_32K	Off
Peripherals	Available	Available	Off	Off	Off
PSRAM	Available	Available	Available	Off	Off
Wakeup on RTC	Available	Available	Available	Off	Off
Wakeup on Pin Edge	Available	Available	Available	Available	Off
Wakeup on BLE	Available	Available	Available	Off	Off
Wakeup on USB	Available	Available	Available	Off	Off
Wakeup on Reset Pin	Available	Available	Available	Available	Available
Brown Out Detector (BOD Interrupt)	Active	Active	Off	Off	N/A
Power On Reset (POR)	Active	Active	Active	Active	N/A

4.1.1.1 Active Mode

The active mode is the default state after power-on reset. In active mode, the Cortex®-M4F is actively executing code. Active mode supports normal operations of the processor and all of the peripherals that are currently enabled. The system clock can be any available clock source (see Table 4-1). The power management by the PMU in the active mode is shown in Figure 4-3.

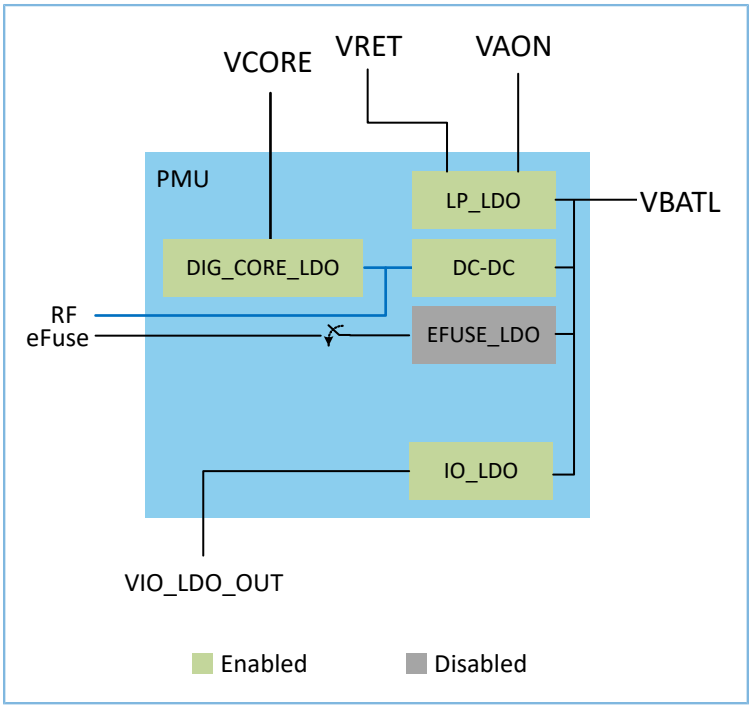

Figure 4-3 Power management in active mode

4.1.1.2 IDLE Mode

In IDLE mode, all unused peripherals can be clocked and no code is executed. The power supply is still applied to the Cortex®-M4F core logic so that it can immediately become active on a wakeup event and begin executing instructions. The power management by the PMU in the IDLE mode is similar to that in the active mode, as shown in Figure 4-3.

To transition from the Active Mode to IDLE Mode, a specific sequence of instructions is executed on the Cortex®-M4F core. First, specific bits in the Cortex®-M4F System Control Register must be set to determine the mode to enter. Refer to *The Definitive Guide to Arm Cortex-M3 and Cortex-M4 Processors*.

The Cortex®-M4F core can enter the IDLE state using one of the following 2 methods:

- Execute a Wait-For-Interrupt (WFI) instruction.
- Execute a Wait-For-Event (WFE) instruction.

Some events will bring the processor back into active mode. The possible reasons to return to Active Mode are:

- A reset.
- An enabled Interrupt is received by the NVIC.
- An event is received by the NVIC.
- A Debug Event is received from the DAP.

4.1.1.3 Deep Sleep Mode

In deep sleep mode, only the always-on (AON) power domain is active. A part of the memory unit is used as retention for storing data that needs to be retained during sleep. Both AON power and retention memory power are supplied from LP_LDO. The power management by the PMU in the deep sleep mode is shown in Figure 4-4.

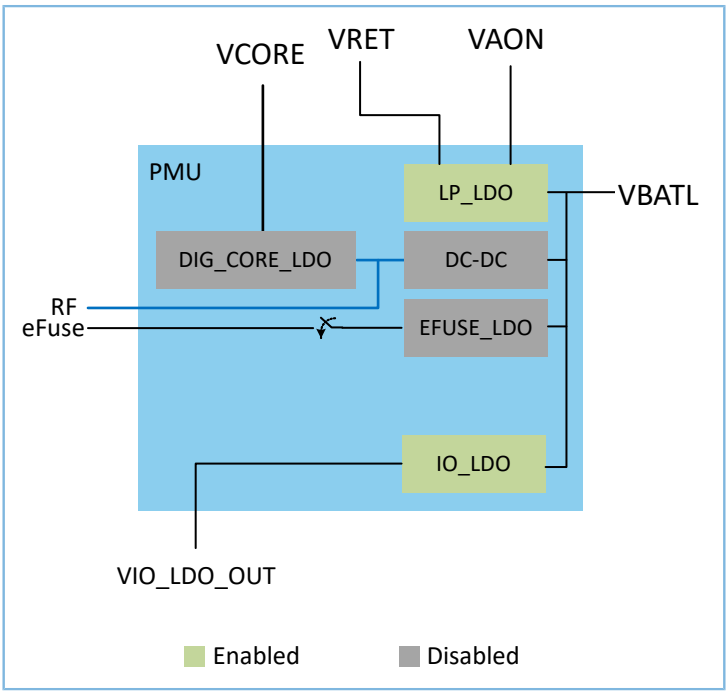

Figure 4-4 Power management in deep sleep mode

The sleep request can be initiated by configuring AON_SLP_CTRL. However, before going to sleep, it is necessary to ensure that the device is not in busy state and the remaining sleep time of the system is longer than the sleep threshold. The integrated GR5526 SDK implements pre-sleep state checks and automatic backup of peripheral configurations that need to be preserved before sleep. I/O and MCU peripherals with retention do not need to be reconfigured when being woken up again, and the CPU that continues execution from warm boot goes into active mode.

The I/Os are latched with the value they are in before entering deep sleep mode. This configuration can be achieved by controlling the DPAD_LE_CTRL register. When DPAD_LE_CTRL.DPAD_LE_SLP_VAL = 0 and DPAD_LE_CTRL.DPAD_LE_WKUP_VAL = 1, the I/Os are latched during sleep and automatically unlocked after waking up. If DPAD_LE_CTRL.DPAD_LE_WKUP_VAL = 0, the I/Os can manually be unlocked after waking up by writing ‘1’ to DPAD_LE_CTRL.DPAD_LE_WR_VAL and DPAD_LE_CTRL.DPAD_LE_APPLY in order. This function has been implemented in the SDK. The default configuration in the SDK is DPAD_LE_CTRL.DPAD_LE_SLP_VAL = 0 and DPAD_LE_CTRL.DPAD_LE_WKUP_VAL = 0.

An external wakeup event, RTC Event or BLE Event, is required to bring the SoC back to active mode. For detailed description of the wake-up source, refer to “Section 4.1.3 Wakeup Source”.

The Cortex®-M4F processor provides a register called the System Control Register (SCR) to allow developers to select between the IDLE mode and deep sleep mode. This register is at an address of 0xE000ED10, and can be accessed in C programming using the “SCB->SCR” symbol. However, the deep sleep mode implementation of the integrated GR5526 SDK has nothing to do with this register. Arbitrary configuration of this register will cause unexpected errors.

4.1.1.3.1 AON_SLP_CTRL

- **Address**: 0xA000A43C
- **Reset Value**: 0x00000000

Table 4-2 AON_SLP_CTRL Register

Bits	Field Name	RW	Reset	Description
31:9	RSVD	R		Reserved bits
8	SLP_DEEP_IN_PROGS	R	0x0	Power controller is processing deep-sleep request.
7:1	RSVD	R		Reserved bits
0	SLP_DEEP_REQ	W	0x0	Write '1' to send sleep request.

4.1.1.3.2 DPAD_LE_CTRL

- **Address**: 0xA000A428
- **Reset Value**: 0x00040000

Table 4-3 DPAD_LE_CTRL Register

Bits	Field Name	RW	Reset	Description
31:19	RSVD	R		Reserved bits
18	DPAD_LE_WKUP_VAL	RW	0x1	Set HW controlled DPAD_LE value after waking up. **Value:** • 0x1: The DPAD latch is transparent. • 0x0: Latches the pad configurations.
17	RSVD	R		Reserved bit
16	DPAD_LE_SLP_VAL	RW	0x0	Set HW controlled DPAD_LE value during sleep. **Value:** • 0x1: The DPAD latch is transparent. • 0x0: Latches the pad configurations.
15:11	RSVD	R		Reserved bits
10	DPAD_LE_CFG_BUSY	R	0x0	DPAD_LE is applying value, do not apply again.
9	RSVD	R		Reserved bit
8	DPAD_LE_RD_VAL	R	0x0	Read the value of DPAD_LE.
7:3	RSVD	R		Reserved bits
2	DPAD_LE_APPLY	W	0x0	Write '1' to apply the value of DPAD_LE_WR_VAL to DPAD_LE control signal.
1	RSVD	R		Reserved bit

Bits	Field Name	RW	Reset	Description
0	DPAD_LE_WR_VAL	RW	0x0	Write the value of DPAD_LE.

4.1.1.4 Ultra Deep sleep mode

Ultra deep sleep mode is the deepest power saving mode that the system can enter. In ultra deep sleep mode, only the always-on (AON) power domain is active. All SRAMs are powered off, so the system restarts execution from cold boot when CPU is woken up from ultra deep sleep mode. The implementation of the system entering ultra deep sleep mode is similar to that in deep sleep mode. The power management by the PMU in ultra deep sleep mode is also similar to that in deep sleep mode, as shown in Figure 4-4.

The I/Os are latched with the value they are in before entering ultra deep sleep mode. This configuration is implemented in the same way as the deep sleep mode.

This mode will block wake-up sources other than AON GPIO peripheral and Sleep timer before going to sleep. Only unmasked wakeup sources can bring the SoC back to active mode. For detailed description of wake-up sources, refer to "Section 4.1.3 Wakeup Source".

4.1.2 SRAM Power Control

The current system supports three SRAM power modes:

- Full power

  SRAM can provide full-speed read-write function. This mode is only used when MCU is powered on.

- Retention power

  When the system enters the sleep mode, the SRAM can enter the retention state during which data can be retained. Read and write operations are not available in this mode.

- Power off

  In this mode, all data will be cleared, and the SRAM will enter the power off state. If users need to control the power supply of SRAM, it is recommended to use the standard functions in the SDK. The SDK also provides the automatic management function of SRAM. This function will be automatically managed according to the RAM used by user programs.

The main Static Random Access Memory (SRAM) is comprised of an on-chip static RAM totaling up to 512 KB. The main SRAM is implemented as several SRAM instances to allow for more control of power usage when less SRAM is required (4 x 32 KB instances, 2 x 64 KB instances, 2 x 128 KB instances). Developers can turn off the redundant SRAM instances or set the active SRAM instances as retention (cannot write or read) in deep sleep mode in order to reduce the power consumption of the SRAM during sleep.

There are 11 power domains for the SRAM, from pd_mcu_00 to pd_mcu_10. The relationship between the SRAM instances and the power domains is shown in Table 4-4.

Table 4-4 SRAM power management

Instance	Access Offset		Size (KB)	Power Domain	Cell Size (KB)
	Low	High			
SRAM0	0x00000000	0x00007FFF	32	pd_mcu_00	16
				pd_mcu_01	16
SRAM1	0x00008000	0x0000FFFF	32	pd_mcu_02	32
SRAM2	0x00010000	0x0001FFFF	64	pd_mcu_03	32
SRAM3	0x00020000	0x0002FFFF	64	pd_mcu_04	64
SRAM4	0x00030000	0x0004FFFF	128	pd_mcu_05	64
				pd_mcu_06	64
SRAM5	0x00050000	0x0006FFFF	128	pd_mcu_07	64
				pd_mcu_08	64
SRAM6	0x00070000	0x00077FFF	32	pd_mcu_09	64
SRAM7	0x00078000	0x0007FFFF	32	pd_mcu_10	32

The SRAM power domain in the wake-up phase can be set through MEM_PWR_WKUP0. The SRAM power domain in the sleep phase can be set through MEM_PWR_SLP0. MEM_PWR_APPLY is used to apply the configuration of MEM_PWR_WKUP0 or MEM_PWR_SLP0. The current working state of the SRAM power domain can be obtained by reading MEM_PWR_STAT0. However, it cannot be read when MEM_PWR_APPLY is busy.

Since the SRAM instances and the power domains do not correspond one-to-one, developers need to configure specific registers according to their relationship shown in Table 4-4. Normally, the first 24 KB of the SRAM0 instance is reserved for Bootloader and BLE protocol stack. Therefore, pd_mcu_00 and pd_mcu_01 should be set in full power in the wake-up phase, and should be set in retention in the sleep phase. For detailed instructions on usage of SRAM, refer to *GR5526 Developer Guide*.

4.1.2.1 Registers

4.1.2.1.1 MEM_PWR_SLP0

- **Address**: 0xA000A480
- **Reset Value**: 0x003FFFFF

Table 4-5 MEM_PWR_SLP0 Register

Bits	Field Name	RW	Reset	Description
31:22	RSVD	R		Reserved bits
21:20	PD_MCU_10_SLP_PWR_SET	RW	0x3	Memory power states that be loaded in sleep phase **Value:** • 0x0: Off

Bits	Field Name	RW	Reset	Description
				• 0x1: Reserved • 0x2: Full power • 0x3: Retention
19:18	PD_MCU_9_SLP_PWR_SET	RW	0x3	Memory power states that be loaded in sleep phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
17:16	PD_MCU_8_SLP_PWR_SET	RW	0x3	Memory power states that be loaded in sleep phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
15:14	PD_MCU_7_SLP_PWR_SET	RW	0x3	Memory power states that be loaded in sleep phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
13:12	PD_MCU_6_SLP_PWR_SET	RW	0x3	Memory power states that be loaded in sleep phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
11:10	PD_MCU_5_SLP_PWR_SET	RW	0x3	Memory power states that be loaded in sleep phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
9:8	PD_MCU_4_SLP_PWR_SET	RW	0x3	Memory power states that be loaded in sleep phase

Bits	Field Name	RW	Reset	Description
				Value: • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
7:6	PD_MCU_3_SLP_PWR_SET	RW	0x3	Memory power states that be loaded in sleep phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
5:4	PD_MCU_2_SLP_PWR_SET	RW	0x3	Memory power states that be loaded in sleep phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
3:2	PD_MCU_1_SLP_PWR_SET	RW	0x3	Memory power states that be loaded in sleep phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
1:0	PD_MCU_0_SLP_PWR_SET	RW	0x3	Memory power states that be loaded in sleep phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention

4.1.2.1.2 MEM_PWR_WKUP0

- **Address**: 0xA000A488
- **Reset Value**: 0x002AAAAA

Table 4-6 MEM_PWR_WKUP0 Register

Bits	Field Name	RW	Reset	Description
31:22	RSVD	R		Reserved bits
21:20	PD_MCU_10_WKUP_PWR_SET	RW	0x2	Memory power states that be loaded in wakeup phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
19:18	PD_MCU_9_WKUP_PWR_SET	RW	0x2	Memory power states that be loaded in wakeup phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
17:16	PD_MCU_8_WKUP_PWR_SET	RW	0x2	Memory power states that be loaded in wakeup phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
15:14	PD_MCU_7_WKUP_PWR_SET	RW	0x2	Memory power states that be loaded in wakeup phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
13:12	PD_MCU_6_WKUP_PWR_SET	RW	0x2	Memory power states that be loaded in wakeup phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
11:10	PD_MCU_5_WKUP_PWR_SET	RW	0x2	Memory power states that be loaded in wakeup phase

Bits	Field Name	RW	Reset	Description
				Value: • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
9:8	PD_MCU_4_WKUP_PWR_SET	RW	0x2	Memory power states that be loaded in wakeup phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
7:6	PD_MCU_3_WKUP_PWR_SET	RW	0x2	Memory power states that be loaded in wakeup phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
5:4	PD_MCU_2_WKUP_PWR_SET	RW	0x2	Memory power states that be loaded in wakeup phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
3:2	PD_MCU_1_WKUP_PWR_SET	RW	0x2	Memory power states that be loaded in wakeup phase **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
1:0	PD_MCU_0_WKUP_PWR_SET	RW	0x2	Memory power states that be loaded in wakeup phase **Value:** • 0x0: Off • 0x1: Reserved

Bits	Field Name	RW	Reset	Description
				• 0x2: Full power • 0x3: Retention

4.1.2.1.3 MEM_PWR_APPLY

- **Address**: 0xA000A490
- **Reset Value**: 0x00000000

Table 4-7 MEM_PWR_APPLY Register

Bits	Field Name	RW	Reset	Description
31:17	RSVD	R		Reserved bits
16	MEM_PWR_APPLY_BUSY	R	0x0	When the bit is set to 1, writing MEM_PWR_APPLY_MAN would not take any effect.
15:1	RSVD	R		Reserved bits
0	MEM_PWR_APPLY_MAN	W	0x0	Write 1 to apply the memory settings manually.

4.1.2.1.4 MEM_PWR_STAT0

- **Address**: 0xA000A498
- **Reset Value**: 0x00000000

Table 4-8 MEM_PWR_STAT0 Register

Bits	Field Name	RW	Reset	Description
31:22	RSVD	R		Reserved bits
21:20	PD_MCU_10_PWR_STAT	R	0x0	Memory power states. Do NOT read when MEM_PWR_APPLY is busy. **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
19:18	PD_MCU_9_PWR_STAT	R	0x0	Memory power states. Do NOT read when MEM_PWR_APPLY is busy. **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power

Bits	Field Name	RW	Reset	Description
				• 0x3: Retention
17:16	PD_MCU_8_PWR_STAT	R	0x0	Memory power states. Do NOT read when MEM_PWR_APPLY is busy. **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
15:14	PD_MCU_7_PWR_STAT	R	0x0	Memory power states. Do NOT read when MEM_PWR_APPLY is busy. **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
13:12	PD_MCU_6_PWR_STAT	R	0x0	Memory power states. Do NOT read when MEM_PWR_APPLY is busy. **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
11:10	PD_MCU_5_PWR_STAT	R	0x0	Memory power states. Do NOT read when MEM_PWR_APPLY is busy. **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
9:8	PD_MCU_4_PWR_STAT	R	0x0	Memory power states. Do NOT read when MEM_PWR_APPLY is busy. **Value:** • 0x0: Off • 0x1: Reserved

Bits	Field Name	RW	Reset	Description
				• 0x2: Full power • 0x3: Retention
7:6	PD_MCU_3_PWR_STAT	R	0x0	Memory power states. Do NOT read when MEM_PWR_APPLY is busy. **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
5:4	PD_MCU_2_PWR_STAT	R	0x0	Memory power states. Do NOT read when MEM_PWR_APPLY is busy. **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
3:2	PD_MCU_1_PWR_STAT	R	0x0	Memory power states. Do NOT read when MEM_PWR_APPLY is busy. **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention
1:0	PD_MCU_0_PWR_STAT	R	0x0	Memory power states. Do NOT read when MEM_PWR_APPLY is busy. **Value:** • 0x0: Off • 0x1: Reserved • 0x2: Full power • 0x3: Retention

4.1.3 Wakeup Source

In order to reduce power consumption, the system will enter deep sleep mode when it is IDLE. In sleep mode, the MCU will be powered off and only the functional modules in the AON area work. In this case, the GR5526 SoC needs a wakeup source to wake the system up when it needs to work.

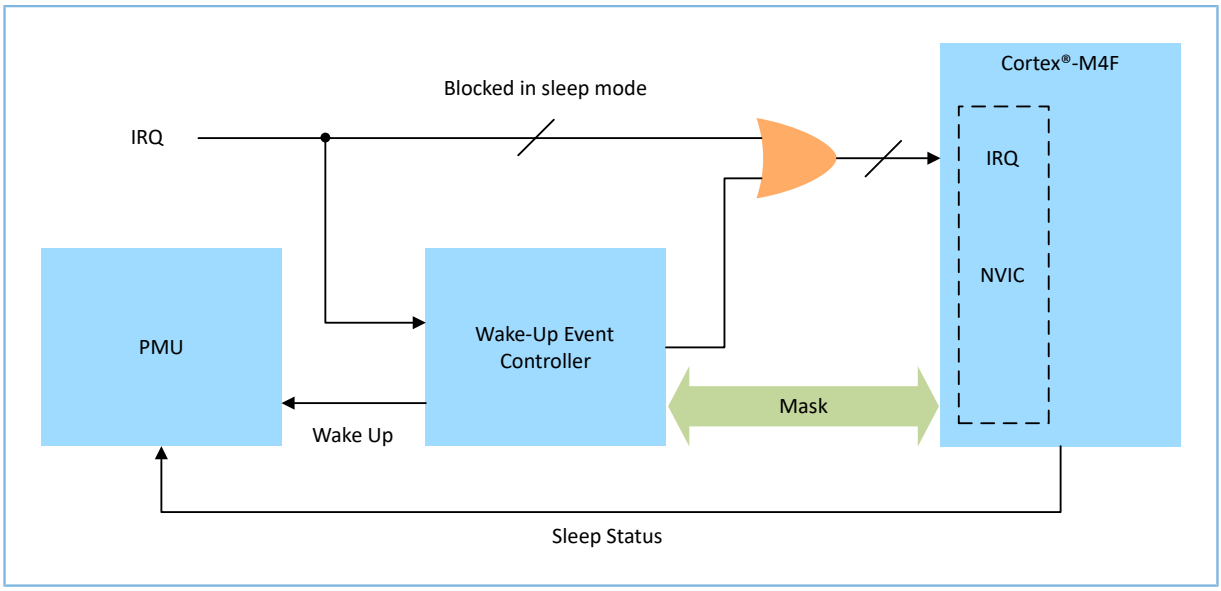

Figure 4-5 Wake-up control block diagram

Currently the SoC supports multiple wakeup sources to wake the system up from deep sleep. Each wakeup source belongs to a different function and is independent of each other.

In deep sleep mode, the device can be woken up through one of the following modules:

- Bluetooth LE controller
- LPCOMP module
- RTC counter
- AON WDT
- Sleep timer
- AON GPIO peripheral
- BOD module
- USB device
- A reset

In ultra deep sleep mode, the device can be woken up through one of the following modules:

- AON GPIO peripheral
- Sleep timer
- A reset

The system is reset when being woken up from the ultra deep sleep mode.

The wakeup source can be enabled by configuring MCU_WAKEUP_CTRL. This wakeup source will wake the MCU up as a wakeup event after the MCU enters the sleep mode. By reading AON_SLP_EVENT, developers can find which wakeup source triggered the wakeup event. It is worth noting that AON_SLP_EVENT needs to be cleared before entering sleep; otherwise the MCU cannot enter sleep mode.

4.1.3.1 MCU_WAKEUP_CTRL

- **Base Address**: 0x4000A080
- **Offset**: 0x00
- **Reset Value**: 0x0000FFFF

Note:

No IRQ is generated if the corresponding source is disabled.

Table 4-9 MCU_WAKEUP_CTRL Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R	0x0	Reserved bits
15:12	RSVD	R	0x0	Reserved bits
11	COMP_F	WR	0x1	PMU_MSIO_COMP_FALLING
10	BLE_IRQ2	WR	0x1	BLE_MAC_IRQ
9	RTC_M2	WR	0x1	RTC_M2, RTC COMPARE, tick event, overflow IRQ
8	USB_DETACH	WR	0x1	USB_DETACH
7	USB_ATTACH	WR	0x1	USB_ATTACH
6	WDT_COMPARE	WR	0x1	WD_TIMER_ALMOST_EXP
5	COMP_R	WR	0x1	PMU_MSIO_COMP_RISING
4	BOD	WR	0x1	PMU_BOD_FEDGE
3	RTC_M1	RW	0x1	RTC_M1, RTC COMPARE, tick event, overflow IRQ
2	BLE	RW	0x1	BLE_TIMER
1	EXT	RW	0x1	EXT_WAKEUP
0	SLP_TIMER	RW	0x1	SLP_TIMER_EXP

4.1.3.2 AON_SLP_EVENT

- **Base Address**: 0x4000A084
- **Offset**: 0x00
- **Reset Value**: 0x00000000

Note:

- Events that happen in sleep or active mode are controlled by MCU_WAKEUP_CTRL.
- To clear an individual event, do not use the conventional read-mask-write method, but simply write ~ (1<< EVENT_BIT) to AON_SLP_EVENT.

Table 4-10 AON_SLP_EVENT Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R	0x0	Reserved bits
15:12	RSVD	R	0x0	Reserved bits
11	COMP_F	WR	0x0	PMU_MSIO_COMP_FALLING
10	BLE_IRQ2	WR	0x0	BLE_MAC_IRQ
9	RTC_M2	WR	0x0	RTC_M2, RTC COMPARE, tick event, overflow IRQ
8	USB_DETACH	WR	0x0	USB_DETACH
7	USB_ATTACH	WR	0x0	USB_ATTACH
6	WDT_COMPARE	WR	0x0	WD_TIMER_ALMOST_EXP
5	COMP_R	WR	0x0	PMU_MSIO_COMP_RISING
4	BOD	WR	0x0	PMU_BOD_FEDGE
3	RTC_M1	RW	0x0	RTC_M1, RTC COMPARE, tick event, overflow IRQ
2	BLE	RW	0x0	BLE_TIMER
1	EXT	RW	0x0	EXT_WAKEUP
0	SLP_TIMER	RW	0x0	SLP_TIMER_EXP

4.2 DC-DC Converter

The DC-DC buck converter supplies power to all RF blocks, digital logic blocks, and memories.

Table 4-11 DC-DC converter specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit	Remarks
Output Voltage	V_{DCDC}	0.9	1.05	1.35	V	Programmable
Load Current	I_{LOAD}			45	mA	
Startup Time	$T_{STARTUP}$		100		µs	
Voltage Ripples	V_{RIPP}		10		mV	Peak-to-peak ripple voltage
Conversion Efficiency	η_{DCDC}		85%			VBATL=3.3 V, I_{LOAD}=10 mA

Note:

Voltage Ripples is obtained based on L_{ext} = 2.2 µH + 9.1 nH and C_{ext} = 2.2 µF

4.2.1 RF_REG_4

- **Address**: 0xA000A814
- **Reset Value**: 0x0098877D

Table 4-12 RF_REG_4 Register

Bits	Field Name	RW	Reset	Description
23	EN_BLEED	RW	0x1	Enables Bleed Current to improve stability
22	LDO_BYPASS	RW	0x0	Enables Bypass mode.
21	CTRL_GATE_ON	RW	0x0	**Value:** • 0x0: Power device is half. • 0x1: Doubles the power device for larger current capability.
20:16	VREF_CTRL_4_0	RW	0x18	The dcdc_vref value to PMU.

4.3 DIG_CORE_LDO

DIG_CORE_LDO regulates the DC-DC converter to supply power to all the digital logic and memory blocks.

Table 4-13 Digital core LDO specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit	Remarks
Input Voltage	V_{in}	0.9	1.05	1.3	V	
Output Voltage	V_{core}	0.85	0.95	1.3	V	Programmable
External Load Cap	C_{load}	0.1	1		µF	

4.3.1 PMU_DCORE_VREF

- **Address**: 0xA000A898
- **Reset Value**: 0x00001401

Table 4-14 PMU_DCORE_VREF Register

Bits	Field Name	RW	Reset	Description
31:13	RSVD	R		Reserved bits
12:8	AON_PMU_DCORE_VREF	RW	0x14	The dcore_vref value to PMU when AON_PUM_DCORE_VREF_SEL = 1
7:1	RSVD	R		Reserved bits
0	AON_PUM_DCORE_VREF_SEL	RW	0x1	**Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0: Use the dcore_vref value from MCU domain. • 0x1: Use the register value.

4.4 IO_LDO

The IO_LDO provides the supply voltage to drive the IO pads.

Table 4-15 IO_LDO specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit	Remarks
Supply Voltage	VBAT	2.4	3.3	4.35	V	
Output Voltage	VIO	1.6	1.8	3.6	V	Programmable; VBAT needs to be greater than VIO + LDO voltage drop.
Quiescent Current	I_q		700		nA	
External Load Cap	C_L	0.1	0.2		µF	
Maximum load current				30	mA	

4.4.1 RF_REG_0

- **Address**: 0xA000A804
- **Reset Value**: 0x070BF470

Table 4-16 RF_REG_0 Register

Bits	Field Name	RW	Reset	Description
31	NC_0	RW	0x0	Reserved
30:24	VREF_SEL_LV_6_0	RW	0x7	Default:1.8 V; LSB = 8.5 mV

4.5 BOD

Brown-out detector (BOD) is a voltage monitoring circuit whose functions include BOD Reset and BOD Interrupt. BOD Reset circuit puts the system into reset state when the supply falls below the Brown-Out Reset threshold. BOD Interrupt circuit is used to indicate the application when the supply voltage drops below the configured interrupt threshold. BOD Interrupt circuit will be prompted in two stages. When the supply voltage drops below the configured interrupt threshold, BOD_ASSERT_IRQn is triggered. When the voltage rises above the sum of the hysteresis voltage and the configured interrupt threshold, BOD_DEASSERT_IRQn is triggered.

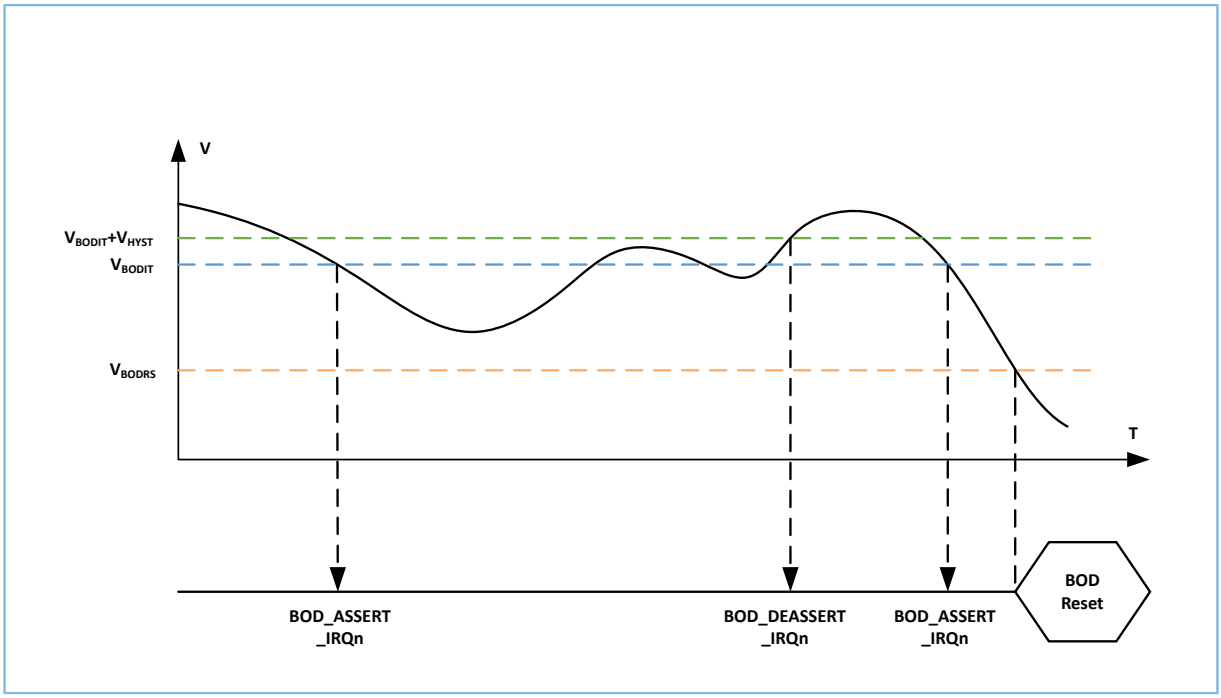

Figure 4-6 BOD Reset and BOD Interrupt

4.5.1 Main Features

- Programmable BOD Interrupt levels
- Hysteresis voltage prevents repeated interruptions.
- Voltage drops and rises with separate interruptions.

4.5.2 Functional Description

In GR5526 SoC, the core of the BOD is the use of two voltage comparators, as Figure 4-7 shows, one for interrupts and one for resets. When the VBATL voltage is lower than the VREF voltage, a falling edge interrupt or reset will be generated. See "Section 8.8.3 Comparator" for more details. The BOD Reset is enabled by default. The BOD Interrupt is optional and the Interrupt threshold level is programmable, so that developers need to configure the BOD Interrupt through the API integrated in the SDK. Developers can also configure the BOD through specific registers.

In sleep mode, BOD Interrupt is not available, while BOD Reset is available in any power mode. See Table 4-1 for more details.

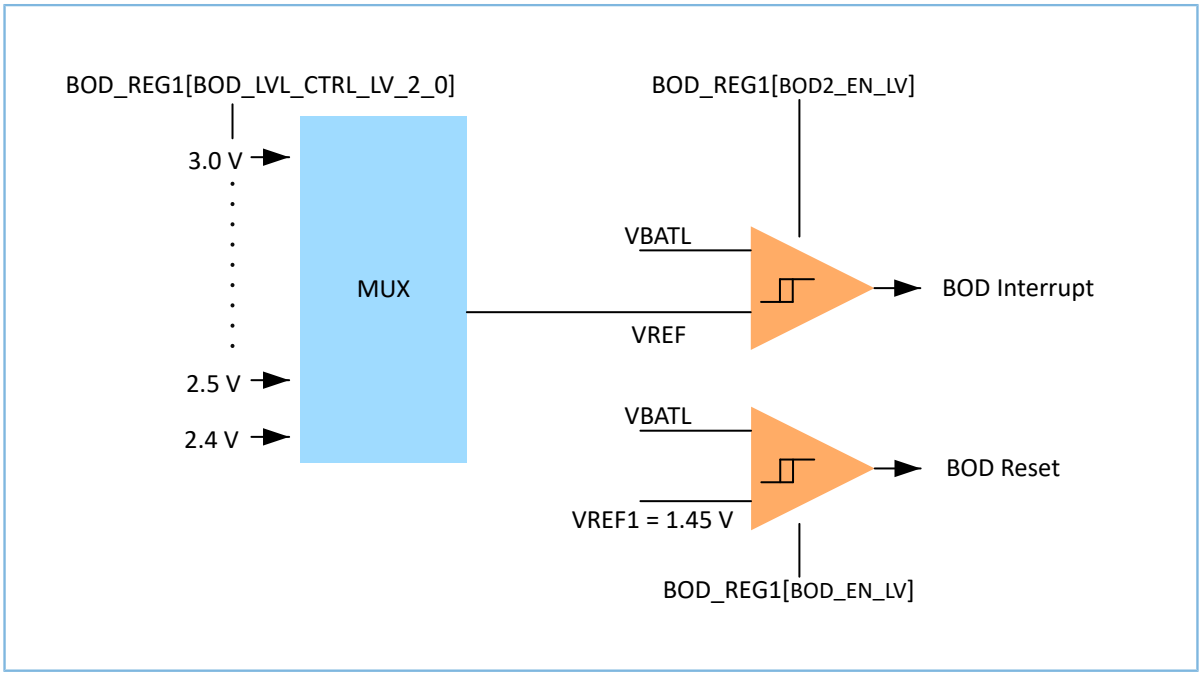

Figure 4-7 Brown-out detector block diagram

The BOD Reset can be enabled by configuring BOD_REG1.BOD_EN_LV. The Brown-Out Reset threshold (V_{BODRS}) is fixed, and its parameters are listed in Table 4-18.

The BOD Interrupt can be enabled by configuring BOD_REG1.BOD2_EN_LV. The interrupt threshold (V_{BODIT}) is a variable to be configured through BOD_REG1.BOD_LVL_CTRL_LV_2_0. A hysteresis voltage (V_{HYST}) of approximately 50 mV prevents voltage fluctuations from causing interrupts to be continuously triggered. Developers also need to configure appropriate interrupt threshold to prevent interrupts from being triggered frequently.

Note that the BOD Interrupt and Reset threshold will not change in different PMU modes.

The BOD Interrupt is also generated when the supply voltage is already below the threshold at the time the BOD Interrupt is enabled, or if the threshold is reconfigured to a higher value than the current voltage, as shown in Figure 4-8. At T = 0, V_{BATL} has already fallen below the V_{BODIT} (for example, when the start-up voltage is stable and V_{BATL} is below the interruption threshold), then BOD_ASSERT_IRQn is generated. The BOD_ASSERT_IRQn will also be generated when the threshold is reset from V_{BODIT} to V_{BODIT2} above V_{BATL}.

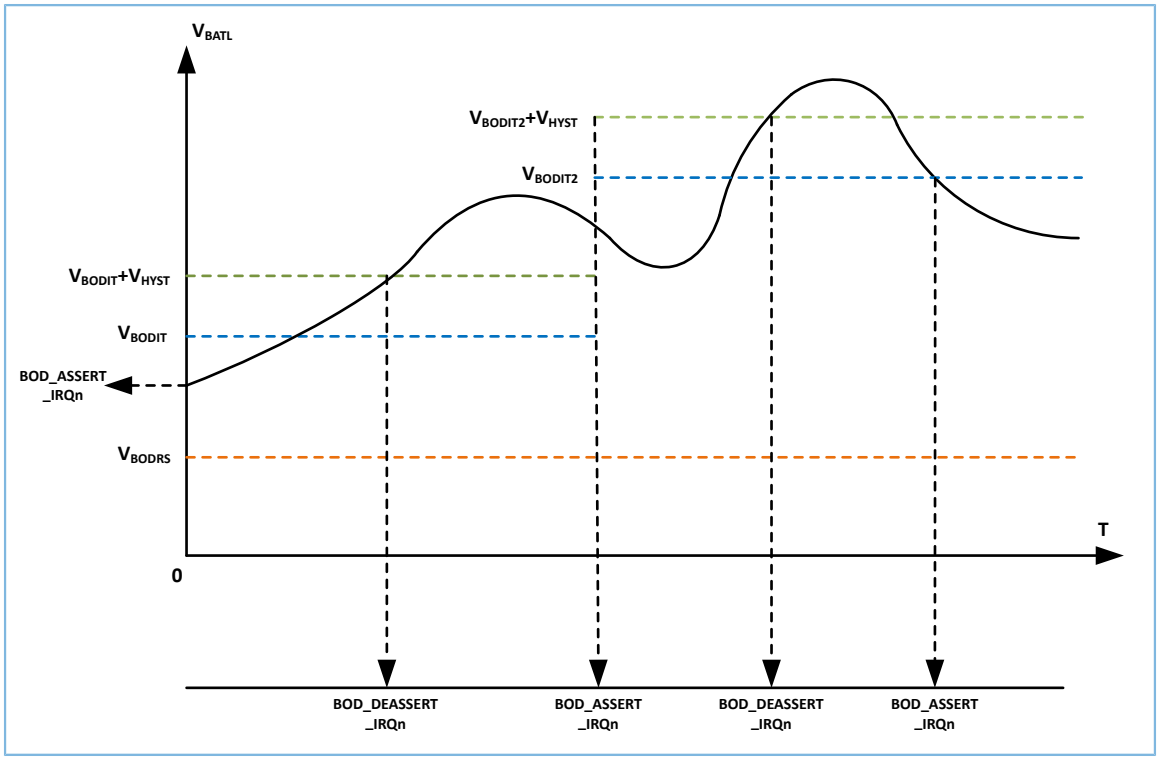

Figure 4-8 BOD Interrupt special case

4.5.3 Registers

4.5.3.1 BOD_REG1

- **Address**: 0xA000A810
- **Reset Value**: 0x00000061

Table 4-17 BOD_REG1 Register

Bits	Field Name	RW	Reset	Description
7	NC_0	RW	0x0	Not connected
6	BOD_MODE_SEL	RW	0x1	**Value**: • 0x1: Enable static mode. • 0x0: Enable dynamic mode. In dynamic mode, the BOD will operate at a certain clock period, with lower power but lower accuracy.
5:2	BOD_INT_LV_SEL	RW	0x8	BOD Interrupt threshold setting; deviation: ±100 mv (depending on temperature and process) **Value**: • 0x9: Set threshold to 2.4 V.

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

Bits	Field Name	RW	Reset	Description
				• 0xA: Set threshold to 2.5 V. • 0xB: Set threshold to 2.6 V. • 0xC: Set threshold to 2.7 V. • 0xD: Set threshold to 2.8 V. • 0xE: Set threshold to 2.9 V. • 0xF: Set threshold to 3.0 V.
1	BOD_INT_EN	RW	0x0	**Value**: • 0x1: Enable BOD interrupt. • 0x0: Disable BOD interrupt.
0	BOD_RESET_EN	RW	0x1	**Value**: • 0x1: Enable BOD reset. • 0x0: Disable BOD reset.

4.5.4 Specifications

In the table below, the typical values are measured at 25 °C, and the minimum and maximum values are measured at the worst process and temperature conditions.

Table 4-18 BOD specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit	Remarks
Brown-Out Reset Threshold	TH_BOD		1.45		V	
Brown-Out Interrupt Threshold	TH_BOD2	1.6		3.0	V	
BOD Response Time (static mode)	RTS_BOD		10		µs	
BOD Response Time (dynamical mode)	RTD_BOD		32		µs	
BOD Hysteresis	HYST_BOD		50		mV	Minimal influence of temperature and process

4.6 Power Consumption

The software constantly adjusts the power supplies of the power management modules (example: DCDC/LDO) according to different temperatures and system clock sources, so estimating the current consumption of an application can be challenging when the measurements cannot be performed directly on different test conditions. To facilitate the estimation, a set of current consumption scenarios are provided to show the typical current drawn from the VDD supply.

Each scenario specifies a set of operations and conditions to apply to the given scenario. All scenarios are listed in Section 4.6.1 Electrical Specifications. The table below shows a set of common conditions used in all scenarios, unless otherwise stated in the description of a given scenario.

Table 4-19 Common conditions

Condition	State	Note
Supply	VBAT = 3.3 V	
Temperature	25°C	
CPU	WFI/WFE/SRPG	
Peripheral	All idle	
Clock	System Clock = HFCLK@96 MHz, Low Clock = LFRC_32K@32.768 kHz	
Flash clock	64 MHz	
Core RAM	Full 512 KB retention	
Cache enabled	Yes	Only applied when the CPU is running from Flash memory.
BLE STACK	Not enabled	
Security Mode	Disabled	
PASRAM	Disabled	
32 MHz crystal	32 MHz, 50 pF ±10 ppm	Only applied when the high-frequency crystal oscillator (HFXO_32M) is running, and HFXO_32M is used when the radio/USB is running.
32 KHz crystal	32.768 KHz, 20 pF ±10 ppm	Only applied when the low-frequency crystal oscillator (LFXO_32K) is running.
BOD/IO_LDO	Enabled/digital IO LDO on	
DCDC/Dig_LDO		
Flash power	Deep sleep mode	
Inductors		
Compiler version	MDK-ARM Plus Version: 5.26.2.0	
Compiler flag	--C99 -c --cpu Cortex-M4F.fp -D__MICROLIB --li -g -O1 --apcs=interwork --split_sections-D__UVISION_VERSION="526"	

4.6.1 Electrical Specifications

4.6.1.1 Sleep

Table 4-20 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
$I_{UDS,RAMOFF,BODOFF,FLASHOFF,RESET}$	Ultra Deep Sleep, no RAM retention, BOD disabled, Flash powered off, woken up on reset		2.38		µA
$I_{UDS,RAMOFF,BODOFF,FLASHOFF,AON_GPIO}$	Ultra Deep Sleep, no RAM retention, BOD disabled, Flash powered off, woken up on AON GPIO		2.38		µA
$I_{UDS,RAMOFF,BODOFF,FLASHOFF,SLP_TIMER}$	Ultra Deep Sleep, no RAM retention, BOD disabled, Flash powered off, woken up on Sleep Timer		2.39		µA
$I_{UDS,RAMOFF,BODOFF,FLASHOFF,SLP_TIMER_LFRC}$	Ultra Deep Sleep, no RAM retention, BOD disabled, Flash powered off, woken up on Sleep Timer (running from LFXO_32K clock)		2.5		µA
$I_{DP,RAMOFF,ANY}$	Deep Sleep, no RAM retention, woken up on any source		2.5		µA
$I_{DP,RAMON,ANY}$	Deep Sleep, full 512 KB RAM retention, woken up on any source		4.3		µA
$I_{DP,RAMON,BLE}$	Deep Sleep, full 512 KB RAM retention, woken up on Bluetooth LE controller		4.2		µA
$I_{DP,RAMON,BLE_LFRC}$	Deep Sleep, full 512 KB RAM retention, woken up on Bluetooth LE controller (running from LFXO_32K clock)		4.4		µA
$I_{DP,128RAMON,BLE}$	Deep Sleep, full 128 KB RAM retention, woken up on Bluetooth LE controller		3.3		µA
$I_{DP,RAMON,SLP_TIMER}$	Deep Sleep, full 512 KB RAM retention, woken up on Sleep Timer		4.2		µA
$I_{DP,RAMON,COMP}$	Deep Sleep, full 512 KB RAM retention, woken up on LPCOMP		4.2		µA
$I_{DP,RAMON,SLP_TIMER_LFXO}$	Deep Sleep, full 512 KB RAM retention, woken up on Sleep Timer (running from LFXO_32K clock)		4.4		µA
$I_{DP,RAMON,RTC}$	Deep Sleep, full 512 KB RAM retention, woken up on RTC		4.4		µA

4.6.1.2 CPU Running

Table 4-21 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
$I_{COREMARK,FLASH}$	CPU running CoreMark from Flash		4.9		mA
$I_{COREMARK,FLASH,HFXO96M}$	CPU running CoreMark from Flash, System Clock = HFXO@96 MHz		5.1		mA
$I_{COREMARK,FLASH,HFRC16M}$	CPU running CoreMark from Flash, System Clock = HFRC@16 MHz		1.8		mA
$I_{COREMARK,FLASH,SEC}$	CPU running CoreMark from Flash, Enable security mode		5.0		mA
$I_{COREMARK,RAM}$	CPU running CoreMark from RAM		4.4		mA
$I_{COREMARK,RAM_ALIAS}$	CPU running CoreMark from RAM Alias		4.9		mA
$I_{COREMARK,RAM_ALIAS,HFRC16M}$	CPU running CoreMark from RAM Alias, System Clock = HFRC@16 MHz		1.8		
$I_{LOOP,FLASH}$	CPU running while loop from Flash		4.5		mA
$I_{LOOP,FLASH,SEC}$	CPU running while loop from Flash, Enable security mode		4.6		mA
I_{LOOP,RAM_ALIAS}	CPU running while loop from RAM Alias		4.7		mA
$I_{LOOP,RAM_ALIAS,HFRC16M}$	CPU running while loop from RAM Alias, System Clock = HFRC@16 MHz		1.7		mA
$I_{IDLE,FLASH}$	CPU idle from Flash		2.4		mA
$I_{IDLE,FLASH,SEC}$	CPU idle from Flash, Enable security mode		2.4		mA
I_{IDLE,RAM_ALIAS}	CPU idle from RAM Alias		2.3		mA

4.6.1.3 Radio Transmitting/Receiving

Table 4-22 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
$I_{TRANSMIT,0dBm,1Mbps,HFXO96M}$	Radio transmitting @0 dBm output power, 1 Mbps Bluetooth LE, System Clock = HFXO@96 MHz		4.0		mA
$I_{TRANSMIT,7dBm,1Mbps,HFXO96M}$	Radio transmitting @7 dBm output power, 1 Mbps Bluetooth LE, System Clock = HFXO@96 MHz		7.7		mA

Symbol	Description	Min.	Typ.	Max.	Unit
$I_{TRANSMIT,-20dBm,1Mbps,HFXO96M}$	Radio transmitting @-20 dBm output power, 1 Mbps Bluetooth LE, System Clock = HFXO@96 MHz		2.2		mA
$I_{TRANSMIT,0dBm,1Mbps,HFXO16M}$	Radio transmitting @0 dBm output power, 1 Mbps Bluetooth LE, System Clock = HFXO@16 MHz		3.7		mA
$I_{RECEIVE,1Mbps,HFXO96M}$	Radio receiving @ 1 Mbps Bluetooth LE mode, System Clock = HFXO@96 MHz		3.5		mA
$I_{RECEIVE,1Mbps,HFXO16M}$	Radio receiving @ 1 Mbps Bluetooth LE mode, System Clock = HFXO@16 MHz		3.4		mA

4.6.1.4 PSRAM

Table 4-23 Electrical specification

Symbol	Description	Min.	Typ.	Max.	Unit
$I_{RUNNING}$	Read/Write @48 MHz		7		mA
$I_{FULL,STANDBY}$	Full array refresh standby current, slow refresh, without CLK toggling		66		µA
$I_{1/2,STANDBY}$	1/2 array refresh standby current, slow refresh, without CLK toggling		65		µA
$I_{1/4,STANDBY}$	1/4 array refresh standby current, slow refresh, without CLK toggling		64		µA
$I_{1/8,STANDBY}$	1/8 array refresh standby current, slow refresh, without CLK toggling		60		µA
$I_{FULL,HS}$	Full array refresh half sleep current, slow refresh, without CLK toggling		20		µA
$I_{1/2,HS}$	1/2 array refresh half sleep current, slow refresh, without CLK toggling		14		µA
$I_{1/4,HS}$	1/4 array refresh half sleep current, slow refresh, without CLK toggling		11		µA
$I_{1/8,HS}$	1/8 array refresh half sleep current, slow refresh, without CLK toggling		10		µA
I_{DPD}	Reset values and memory content are not retained in Deep Power Down mode.		7		µA

4.6.1.5 USB

Table 4-24 Electrical specification

Symbol	Description	Min.	Typ.	Max.	Unit
$I_{RAMON,RUNNING,USB_ACTIVE}$	Full 512 KB RAM retention, CPU running, USB active				mA
$I_{RAMON,SLEEPING,USB_SUSPEND}$	Full 512 KB RAM retention, CPU sleeping, USB suspended				µA

4.6.1.6 ADC Active

Table 4-25 Electrical specification

Symbol	Description	Min.	Typ.	Max.	Unit
$I_{ADC,4KHz,128_POINTS}$	CPU running, ADC sampling@4 kHz sample clock, acquire 128 points				mA
$I_{ADC,16KHz,128_POINTS}$	CPU running, ADC sampling@16 kHz sample clock, acquire 128 points				mA
$I_{ADC,1MHz,128_POINTS}$	CPU running, ADC sampling@1 MHz sample clock, acquire 128 points				mA
$I_{ADC,16MHz,128_POINTS}$	CPU running, ADC sampling@16 MHz sample clock, acquire 128 points				mA

4.6.1.7 TEMP Measure Active

Table 4-26 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
I_{TEMP}	TEMP measurement started				mA

4.6.1.8 VBATL Measure Active

Table 4-27 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
I_{VBATL}	VBATL measurement started				mA

4.6.1.9 COMP Active

Table 4-28 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
I_{COMP}	COMP started				µA

4.6.1.10 BOD Active

Table 4-29 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
I_{BOD}	BOD running		0.4		µA

4.6.1.11 TRNG Active

Table 4-30 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
I_{TRNG}	TRNG running				µA

4.6.1.12 Security Active

Table 4-31 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
$I_{PRESENT128}$	Present-128 running in encryption mode				µA
I_{AES}	AES running				µA
I_{PKC}	PKC running				µA
I_{HMAC}	HMAC running				µA

5 Reset

5.1 Functional Overview

A reset is triggered by power, hardware, or software in GR5526 SoC. Power reset includes power-on reset and brown-out reset. Hardware reset includes external reset pin and watchdog reset. Software reset refers to the NVIC System Reset Register (AIRCR) as defined in Cortex®-M4F.

The reset structure of GR5526 is illustrated in Figure 5-1.

The PMU, triggered by CHIP_EN pad, POR circuit, or BOD reset circuit, generates an active low por_rst_n reset signal. The watchdog reset by the watchdog timer in the always-on power domain generates an active low aon_wdog_rst_n reset signal. Either of these two reset signals will cause an active low pwr_rst_n reset signal to the power state controller, resulting in generating a reset signal of the SoC.

The power state controller generates an active low MCU Core Reset (mcu_core_rst_n), which goes directly to Cortex®-M4F core reset input. In order to support CPU-initiated system reset, the System Reset Request output of Cortex®-M4F is combined with the mcu_core_rst_n, and the resulting active low System Reset (sys_rst_n) is connected to MCU peripherals as well as the XF Cache and always-on modules.

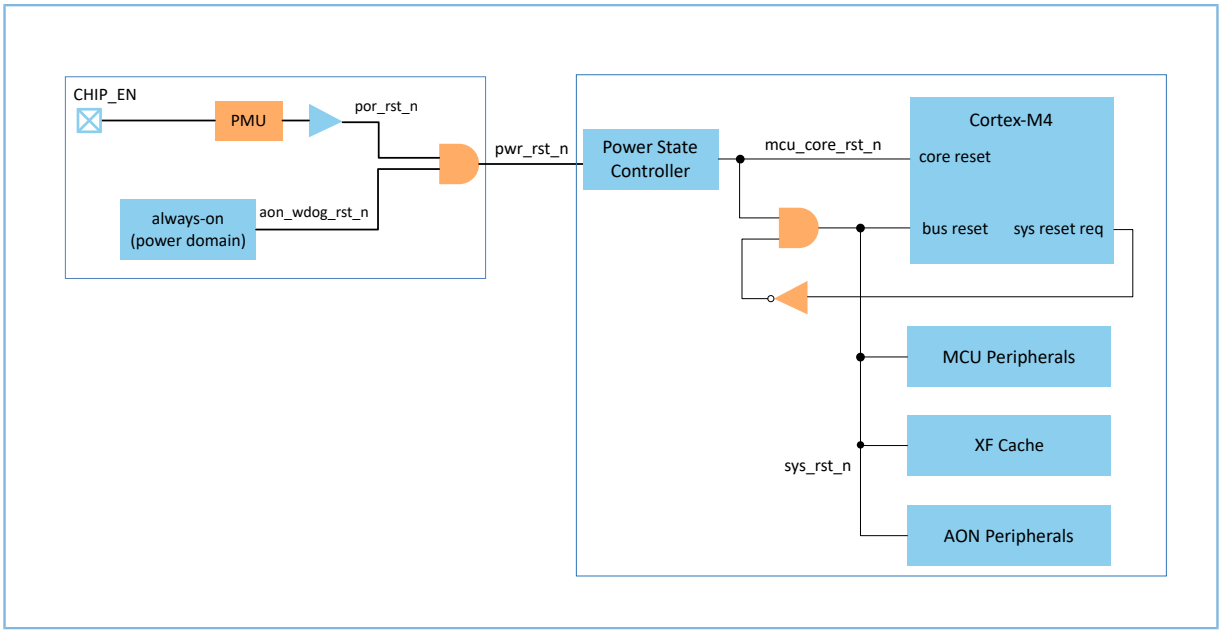

Figure 5-1 Reset structure of GR5526

The reset table is illustrated as below:

Table 5-1 Impact of reset sources on modules

Reset Source	Power Reset		Hardware Reset		Software Reset
	POR	BOD	External Reset Pin	Watchdog	SYSRESETREQ
Power State Controller	Impacted	Impacted	Impacted	Impacted	Not impacted

Reset Source	Power Reset		Hardware Reset		Software Reset
	POR	BOD	External Reset Pin	Watchdog	SYSRESETREQ
Always-on peripherals	Impacted	Impacted	Impacted	Impacted	Impacted
BLE timer	Impacted	Impacted	Impacted	Impacted	Impacted
BLE core	Impacted	Impacted	Impacted	Impacted	Impacted
MCU subsystem	Impacted	Impacted	Impacted	Impacted	Impacted
Cortex®-M4 core	Impacted	Impacted	Impacted	Impacted	Impacted
Cortex®-M4 debug	Impacted	Impacted	Impacted	Impacted	Not impacted

5.2 Power Reset

5.2.1 Power-on Reset

Power-on Reset (POR) initializes the system when the supply voltage is above the power-on threshold. The system will be held in the reset phase until the supply voltage has reached the minimum operating voltage. The power-on threshold V_{work} of the GR5526 SoC is about 1 V.

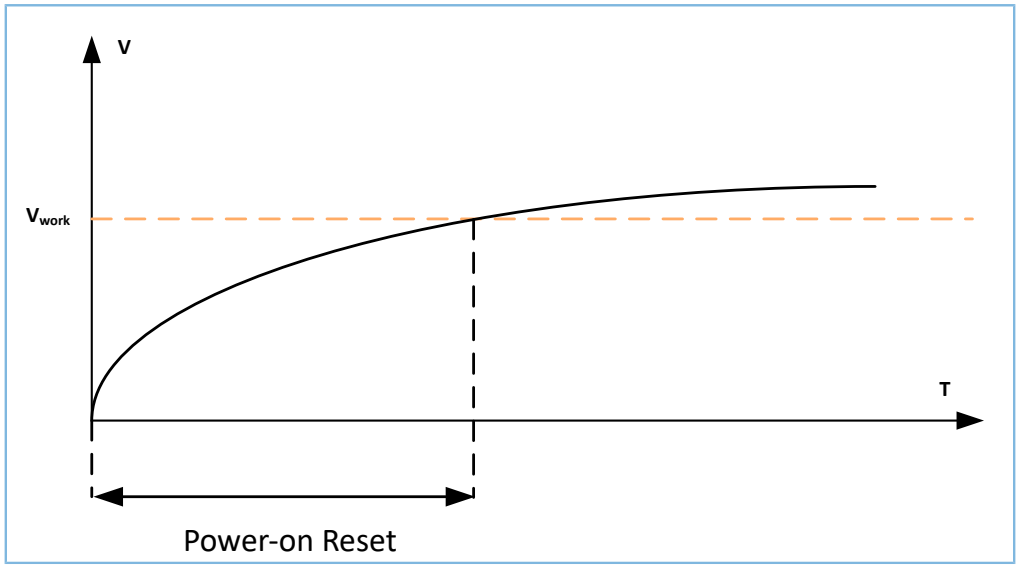

Figure 5-2 Power-on reset

5.2.2 Brown-out Reset

Brown-out Reset is dependent on the Brown-out detector (BOD). BOD Reset circuit puts the system into reset state when the supply falls below the Brown-Out Reset threshold.

For detailed description of the BOD reset, refer to “Section 4.5 BOD”.

5.3 Hardware Reset

5.3.1 External Reset Pin

- The active-low CHIP_EN pin can be used to generate a reset by using an off-chip component.
- There is no internal pull-up resistor on this pin. The CHIP_EN pin is not maskable.

5.3.2 Watchdog Reset

A watchdog timer (WDT) reset is generated when the watchdog timer times out. The watchdog is on by default, and its time-out count value is 200,000 clock cycles of "aon_timer_clock", roughly 6 seconds by default. The software will re-perform cold boot after watchdog reset.

For detailed description of the watchdog timer, refer to "Section 7.1 Always-on Watchdog (AON WDT)".

5.4 Software Reset

A reset may be generated via software by using the NVIC System Reset Register (AIRCR) defined in the Cortex®-M4. A software reset can be generated by controlling the SYSRESETREQ bit of this register. The software will re-perform cold boot after software reset.

For detailed description of the NVIC reset, refer to *The Definitive Guide to Arm Cortex-M3 and Cortex-M4 Processors*.

5.5 Reset Reason Recorder

In a GR5526 SoC, a reset reason recorder module is implemented in the always-on domain. When a reset is triggered, a specific flag register is set by hardware. The flag register is accumulative, which means it holds the status until a software clear is applied. It is recommended to clear the flag before starting recording.

5.5.1 Registers

- **Name:** DBG_REG_RST_SRC
- **Description:** This register controls the reset reason recorder.
- **Base Address:** 0x4000A000
- **Offset:** 0x0384
- **Reset Value:** 0x1000000

Table 5-2 DBG_REG_RST_SRC Register

Bits	Field Name	RW	Reset	Description
31	DBG_RST_SRC_CFG	WO	0x0	1: Write 1 to apply the bit [25:24]
30	DBG_RST_SRC_CFG_BUSY	RO	0x0	1: The block is being configured. Any other write operation is not allowed.
29:26	RSVD			
25	DBG_SRC_CLR	RW	0x0	1: Clear all the status
24	DBG_RST_SRC_EN	RW	0x1	1: Enable reset source recording.

Bits	Field Name	RW	Reset	Description
23:17	RSVD			
16	DBG_RST_SRC_RDY	RO	0x0	• 0: Reset Reason Recorder is not ready to record. • 1: Reset Reason Recorder is ready to record.
15:6	RSVD			
5:0	DBG_RST_SRC_STAT	RO	0x0	Reset season status: • Bit 0 – System reset (SYSRESETREQ) • Bit 1 – Reserved • Bit 2 – AON watchdog reset • Bit 3 –Reserved • Bit 4 --POR • Bit 5 – Reserved

5.5.2 Programming

When the DBG_REG_RST_SRC register is used to clear flags. The following applies:

- Writing “1” to both bit[25] and bit[31] starts a clearing process.
- As clock domain crossing takes some time, perform polling on bit[30] until the Reset Reason Recorder is not busy (low).
- Writing “0” to bit[25] and “1” to bit[31] exits the clearing process.

When the DBG_REG_RST_SRC register is used to enable reset recording, the following applies:

- Writing “1” to both bit[24] and bit[31] starts an enabling process.
- As clock domain crossing takes some time, perform polling on bit[16] until the Reset Reason Recorder block is ready (high).

6 Clocks

6.1 Functional Overview

Clock system is an important structure in the GR5526 SoC. The clock sources can be derived from a series of internal or external high and low frequency oscillators. Different modules can use corresponding clock source according to their own requirements. The clock sources are divided into high-frequency clock and low-frequency clock in the GR5526 SoC as below.

High-frequency clock sources:

- HFXO_32M
- CPLL_192M
- HFRC_192M

Low-frequency clock sources:

- RNG_OSC
- LFXO_32K
- LFRC_32K

Figure 6-1 shows the basic clock structure of the GR5526 SoC.

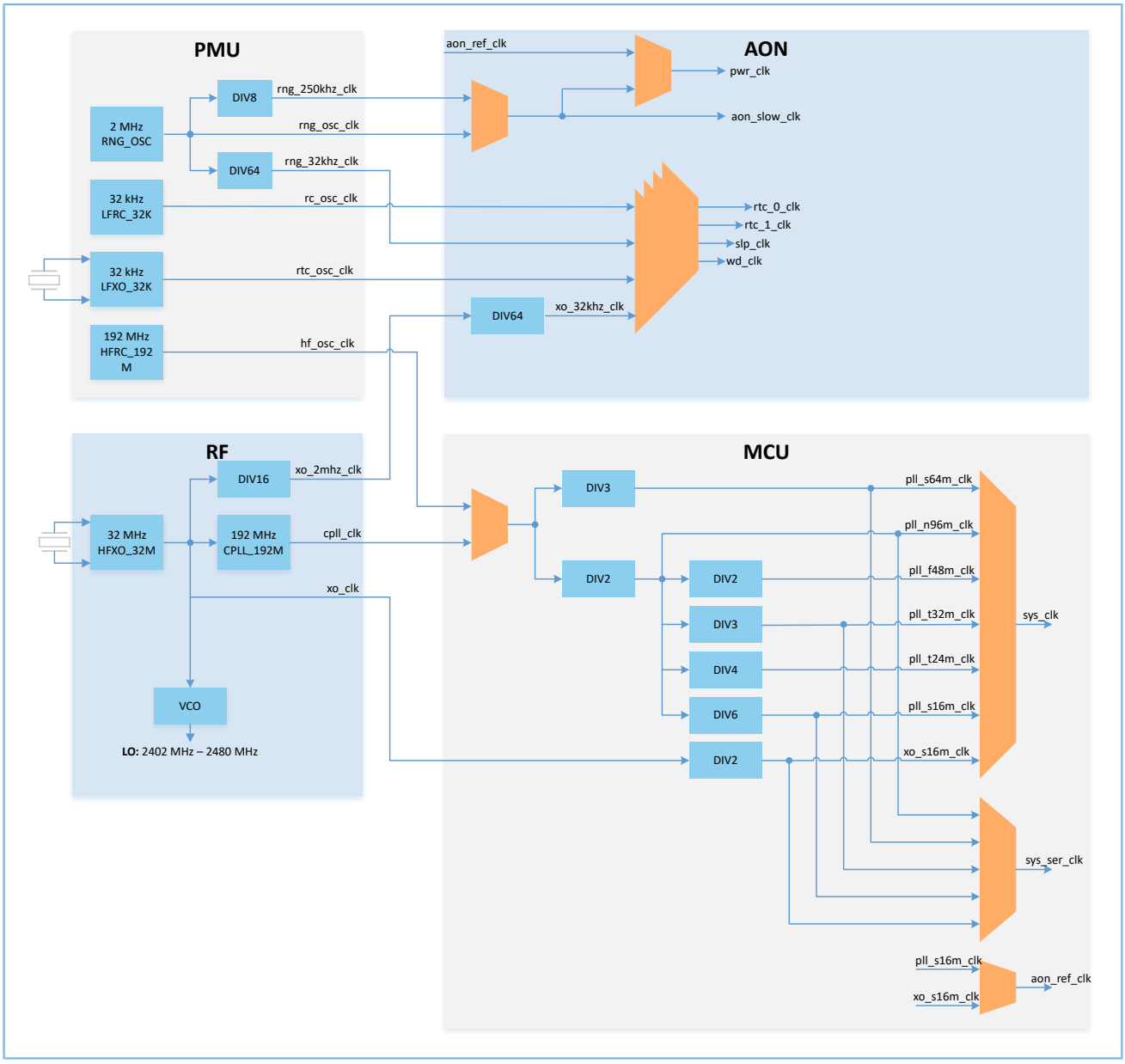

Figure 6-1 MCU clocks

MCU clocks are generated using a high-frequency clock (CPLL_192M) that requires a 32 MHz external crystal (HFXO_32M) or an internal high-frequency RC oscillator (HFRC_192M) when the wireless link is not used. During the MCU working stage, either CPLL_192M or HFRC_192M must be available. The external crystal is needed to meet the offset requirements of the Bluetooth LE specifications.

The system clock and system serial clock can be configured independently. In the GR5526 SDK, the default value of the system serial clock is 96 MHz.

The AON Power State Controller clock comes from a 2 MHz ring oscillator (RNG_OSC) with a choice of divided by 250 kHz clock (rng_250khz_clk) or undivided 2 MHz clock (rng_osc_clk).

The clocks for the various slow AON timer modules (including BLE Timer, Sleep Timer, RTC, and WDT) can be generated from a choice of 4 sources: low-frequency RC oscillator (LFRC_32K), divided-down version of 2 MHz RNG_OSC, low-

frequency crystal (LFXO_32K) based on 32 kHz crystal, and divided-down version of HFXO_32M clock (XO_32KHz_CLK). Due to the dependence between these modules and clock sources, once the clock source is selected for the module, this clock source should not be turned off. Since the clock source of the XO_32KHz_CLK is from HFXO_32M, the average power consumption of the system will increase, and the accuracy depends on the HFXO_32M in this situation.

A divided-down version of the HFXO_32M clock can be muxed out through one of the AON GPIO5 to provide an accurate 2 MHz clock output to an external peripheral.

For XO clock sources such as HFXO_32M and LFXO_32K, GR5526 SoC does not require external capacitors except the crystal itself. It is recommended to choose a crystal with low load capacitance (6 pF–8 pF). Low load capacitance will reduce startup time and current consumption. For specified parameters and recommended selection of external crystals, refer to *GR5526 Hardware Design Guidelines*.

To reduce power consumption, an integer clock supporting Phase-Locked Loop (CPLL_192M) is used to generate the main system clock to both RF and digital sections. Frequency dividers are used to provide slower frequencies for different blocks of GR5526. CPLL_192M is also used to tune the desired frequency hopping for each packet. The settling time of the CPLL_192M is designed to be within the time defined by the Bluetooth LE specifications to maintain the interframe space (IFS) timing.

Table 6-1 Clock sources with power management

Clock Source	Active Mode	Deep Sleep Mode
HFXO_32M	ON (Conditionally OFF)	Conditionally OFF
CPLL_192M	ON (Conditionally OFF)	Normally OFF
HFRC_192M	ON (Conditionally OFF)	Normally OFF
RNG_OSC	ON	ON
LFXO_32K	ON (Conditionally OFF)	ON (Conditionally OFF)
LFRC_32K	ON (Conditionally OFF)	ON (Conditionally OFF)

The usage of each clock source in different power modes is described in Table 6-1. Developers can selectively turn off some clock sources in order to reduce the average power consumption of the system. Some clock sources are turned off conditionally, which can be found in the chapter description of each clock. It's worth noting that when the clock is turned on, it usually takes time for the crystal to start oscillating, and it also takes time for the output to fully stabilize. In order to prevent erroneous clocks, GR5526 SDK implements automatic calculation of the time required for different clocks to stabilize and delays until the clocks are stabilized before being fully activated.

For power optimization, a lot of clock gates (CG) are implemented, as showed in Figure 6-2.

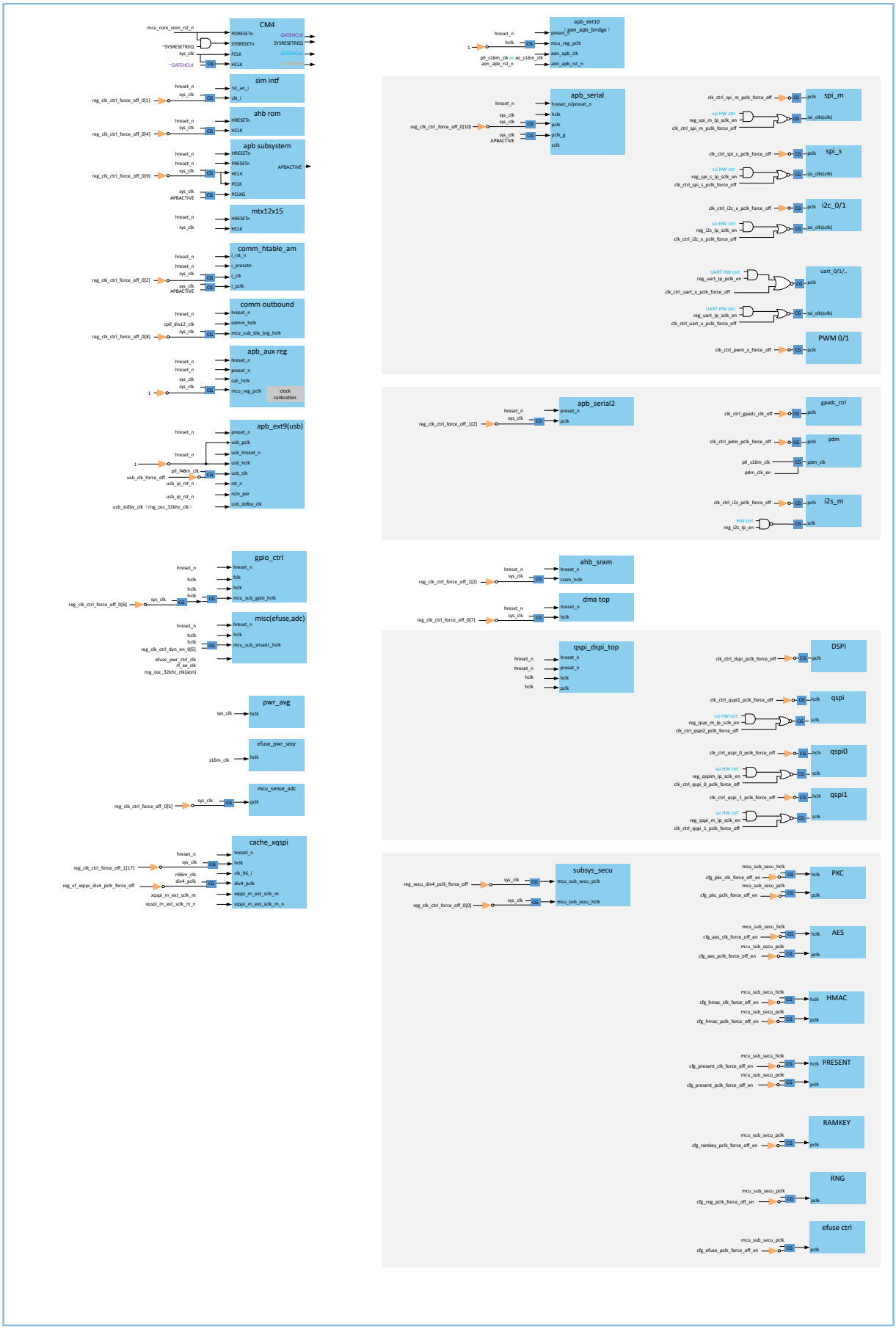

Figure 6-2 Clock gates (CG)

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

The clock of the module can be controlled by the CGs. The CGs can be configured manually or controlled automatically by hardware, which improves the flexibility of power controlling. The clocks of each module can be forced off by controlling the *_force_off registers.

The module driver in the GR5526 SDK has implemented that the corresponding clock is automatically turned on through CGs. Through the initialization API in the SDK, developers do not need to manually open the module clock.

The CG control management in sleep mode has been implemented in the GR5526 SDK. Since the peripheral's control registers are not in the AON power domain, the configuration will be lost after entering sleep mode so that the critical register configuration should been backed up before sleep. The clocks will be turned on through the CGs during the wake-up phase, and the critical register configuration will be restored automatically.

Developers is able to control the clock of each module through the CG register described in “Section 6.5 Registers”.

6.2 High Frequency Clock

6.2.1 HFXO_32M

HFXO_32M clock is a crystal-based oscillator that connects to a 32 MHz external crystal. The startup time of the HFXO_32M clock depends on the crystal type as well as the internal bias current. The bias current is programmable to reduce the power consumption when the HFXO_32M clock settles.

The GR5526 SoC system will use the xo_s16m_clk from HFXO_32M divided by two as the system clock at the initial stage of power-on. After normal booting, the GR5526 SoC is able to switch to user-configured system clock.

HFXO_32M clock is used in application scenarios that require high precision and temperature sensitivity, such as Bluetooth RF and USB. When processing Bluetooth tasks and USB tasks, the HFXO_32M clock should be turned on.

Normally, HFXO_32M can be turned off in sleep mode, which can reduce system power consumption. However, if AON timers (including BLE Timer, Sleep Timer, RTC, and WDT) use xo_32khz as clock source, HFXO_32M cannot be turned off in sleep mode, because xo_32khz is down-frequency generated by HFXO_32M.

Table 6-2 HFXO_32M clock specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit
HFXO_32M frequency	$HFXO_32M_{FREQ}$		32		MHz
Accuracy	$HFXO_32M_{ACC}$			100	ppm

6.2.2 CPLL_192M

The high-speed clock is obtained by multiplying the HFXO_32M clock source by the phase-locked loop (CPLL_192M) circuit. The multiplied clock is used to generate the main system clock for the RF and digital sections.

CPLL_192M is also used to tune the desired frequency hopping for each packet. The settling time of the CPLL_192M is designed to be within the time defined by the Bluetooth LE specifications to maintain the interframe space (IFS) timing.

Table 6-3 CPLL_192M specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit
Output frequency	$CPLL_192M_{FREQ}$		192		MHz
Deviation across Temp	$CPLL_192M_{ACC}$	–500		500	ppm/°C

6.2.3 HFRC_192M

The high-frequency RC oscillator (HFRC_192M) is an internal 192 MHz clock used for fast system start, providing system clock for MCU.

HFRC_192M will be open during the initial stage of power-on by default. Due to the individual differences of the RC oscillator, the frequency of HFRC_192M may be different variable on different SoCs. Developers can adjust the frequency through specific registers to meet the clock frequency requirements of each module.

Due to poor accuracy and temperature sensitivity, the high-frequency RC oscillator (HFRC_192M) is required to switch to HFXO_32M clock if Bluetooth LE connection is used. In applications where no high-precision clock is required, the high-frequency RC oscillator (HFRC_192M) can replace the high-frequency RC oscillator (CPLL_192M) in order to reduce the power consumption.

Table 6-4 HFRC_192M specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit
Output frequency	$HFRC_192M_{FREQ}$	183.3	192	196	MHz
Deviation across temperature	$HFRC_192M_{ACC}$	–500	265	500	ppm

In GR5526 SoC, the actual frequency of HFRC_192M can be controlled by registers. Through the FS_REG_1 register, the actual frequency of HFRC_192M can be roughly adjusted. Through the FS_REG_0 register, the actual frequency of HFRC_192M can be finely adjusted. The HFRC_192M can be calibrated to the ideal frequency through the combination of coarse and fine adjustments. The actual frequency of the current clock can be obtained through the clock calibration method in “Section 6.4 Clock Calibration”.

The integrated GR5526 SDK has implemented the function of automatically calibrating HFRC_192M during startup. Developer can also adjust the frequency of HFRC_192M again through the register to make it reach the desired frequency.

6.2.3.1 FS_REG_0

- **Address**: 0xA000A850
- **Reset Value**: 0x010010C6

Table 6-5 FS_REG_0 Register

Bits	Field Name	RW	Reset	Description
24:16	ctrl_osc_fine_8_0	RW	0x100	Fine code for hf_clk frequency adjustment. Used for background calibration. LSB–1000 ppm.

6.2.3.2 FS_REG_1

- **Address**: 0xA000A854
- **Reset Value**: 0x00005428

Table 6-6 FS_REG_1 Register

Bits	Field Name	RW	Reset	Description
3:0	ctrl_osc_crs_3_0	RW	0x8	Coarse code that controls hf_clk frequency. Default: 1000 (Need one-time calibration using the on-chip frequency calibration)

6.3 Low Frequency Clock

6.3.1 RNG_OSC

The ring oscillator (RNG_OSC) clock is a low-frequency (2 MHz) and ultra-low power clock that is ON as long as the chip is enabled. This clock is used to clock the power sequencers at both the cold and warm boots of the SoC.

Due to high power consumption when the frequency is 2 MHz, in order to reduce the power consumption during sleep, the GR5526 SoC will switch to rng_250khz_clock as the clock source during sleep. In active mode, the GR5526 SoC will switch back to 2 MHz frequency for faster response speed. If developers have no special requirements for the power consumption during sleep, the frequency during sleep can be kept at 2 MHz, which can improve the wake-up speed.

Table 6-7 RNG_OSC specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit
Output frequency	RNG_OSC_{FREQ}		1.3		MHz
Deviation across Temp	RNG_OSC_{ACC}	-500		500	ppm/°C

6.3.2 LFXO_32K

LFXO_32K clock is a crystal-based accurate clock (32.768 kHz) that is more stable than the low-frequency RC oscillator (LFRC_32K) across voltage and temperature. This clock is used for accurate sleep and wakeup.

In some cost-sensitive scenarios, the external crystal can be removed. In this situation, the SoC will not have the clock source of LFXO_32K. Low-frequency RC oscillator (LFRC_32K), divided-down version of 2 MHz RNG_OSC, and divided-down version of HFXO_32M clock are still available for being the clock source.

In sleep mode, LFXO_32K can be turned off unless it is used as the clock source for the AON timers (including BLE Timer, Sleep Timer, RTC, and WDT).

LFXO_32K can be muxed out to AON_GPIO4 in sleep mode.

Table 6-8 LFXO_32K specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit
Output frequency	$LFXO_32K_{FREQ}$		32.768		kHz
Accuracy	$LFXO_32K_{ACC}$			500	ppm

6.3.2.1 RF_REG_0

- **Address**: 0xA000A804
- **Reset Value**: 0x070BF470

Table 6-9 RF_REG_0 Register

Bits	Field Name	RW	Reset	Description
7	rtc_en	RW	0x0	Enable LFXO_32K. **Value:** • 0x0: Disable • 0x1: Enable

6.3.3 LFRC_32K

The low-frequency RC oscillator (LFRC_32K) is an internal 32 kHz clock used for generating various slow AON timer module clocks. Compared with LFXO_32K, LFRC_32K has lower precision.

In applications where no high-precision clock is required, the low-frequency RC oscillator (LFRC_32K) can replace the low-frequency crystal (LFXO_32K).

In sleep mode, LFRC_32K can be turned off unless it is used as the clock source for the AON timers (including BLE Timer, Sleep Timer, RTC, and WDT).

LFRC_32K can be muxed out to AON_GPIO4 in sleep mode.

Table 6-10 LFRC_32K specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit
Output frequency	$LFRC_32K_{FREQ}$	24	33	40	kHz
Deviation across temperature	$LFRC_32K_{ACC}$	–500	123	500	ppm/°C

6.3.3.1 RC_RTC_REG_0

- **Address**: 0xA000A848

- **Reset Value**: 0x6010270B

Table 6-11 RC_RTC_REG_0 Register

Bits	Field Name	RW	Reset	Description
15	rcosc_en	RW	0x0	Enable LFRC_32K. **Value:** • 0x0: Disable • 0x1: Enable

6.4 Clock Calibration

GR5526 SoC provides a clock calibration module. This module calibrates inaccurate clocks by using accurate clocks. One of these two clocks should be accurate, the other is to be calibrated. When using a high-speed clock to calibrate a low-speed clock, the frequency difference between the two clocks should be large enough in order to obtain reasonable calibration results. When calibrating a high-speed clock with a low-speed clock, there are inevitable errors in the result due to frequency limitations.

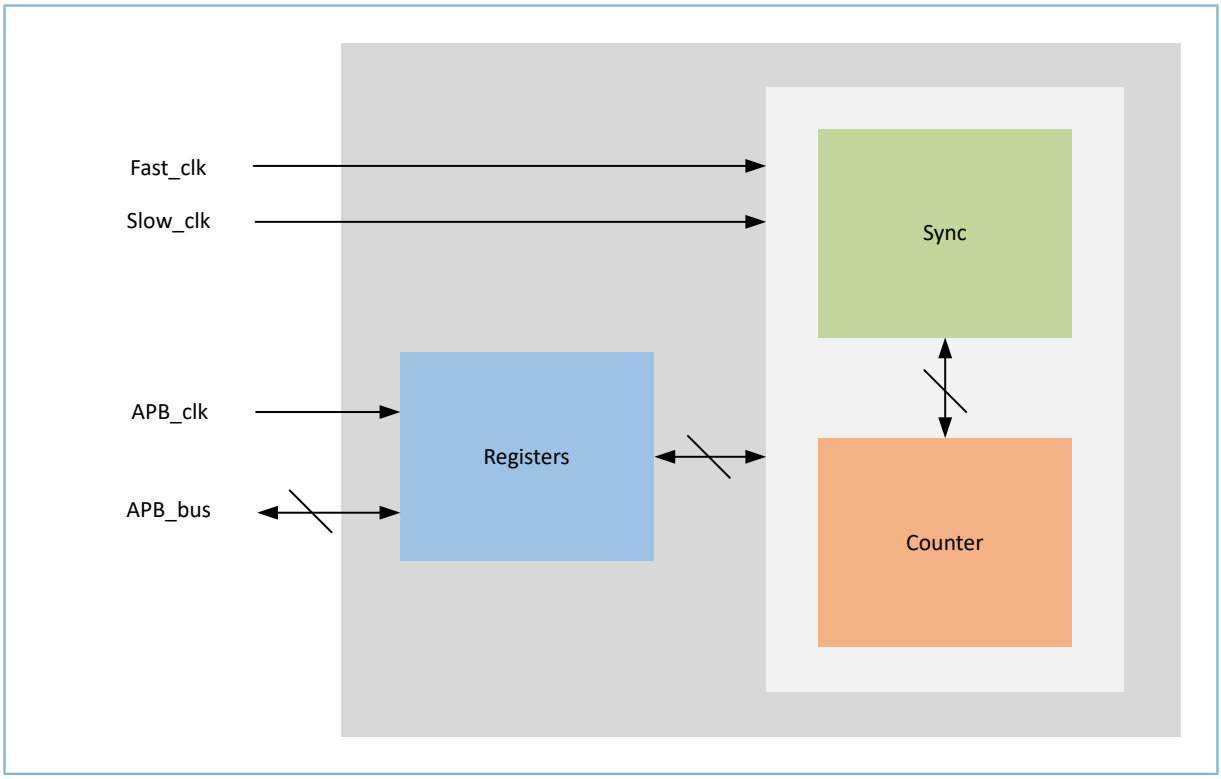

Figure 6-3 Clock calibration block diagram

The fast clock is used to sample the slow clock, or the slow clock is used to sample the fast clock. By reading the result of sampling, the relationship of the two clocks can be calculated. The relationship of the fast clock (f_{fast}) and the slow clock (f_{slow}) can be calculated by the following equation:

$$\frac{f_{slow}}{N_{slow}} = \frac{f_{fast}}{N_{fast}} \quad (2)$$

6.4.1 Operation

GR5526 SoC provides two sets of registers for clock calibration. The SL prefixed registers are used to calibrate the kHz clock (low-frequency clock) such as LFXO_32K, LFRC_32K or RNG_32K from RNG_OSC with high-frequency clock (HFXO_32M). The HS prefixed registers are used to calibrate the HFRC_192M clock with low-frequency clock such as LFXO_32K, LFRC_32K or XO 31.25 kHz from HFXO_32M.

Normally, it is recommended to use XO 31.25 kHz as slow clock source to calibrate HFRC_192M in order to get better calibration results. When using XO 31.25 kHz for calibration, HFXO_32M needs to be turned on. When using LFXO_32K or LFRC_32K as the low-speed clock source to calibrate HFRC_192M, LFXO_32K and LFRC_32K clock sources need to be calibrated by using the HFXO_32M clock sources first, so as to ensure that LFXO_32K or LFRC_32K as the reference clock source is valid.

The following operation flow applies to two sets of registers.

Operation flow:

1. Set the slow clock source (CLK_SEL). SL_CLK_SEL refers to the slow clock to be calibrated. HS_CLK_SEL refers to the slow clock used to be reference.
2. Set the number of clock cycles of the clock to be calibrated (CLK_CNT.cnt_val = x, where final N = x + 1).
3. Enable the calibration module (CLK_CTRL.enable = 1).
4. Wait the module finishing counting (CLK_STAT.ready = 1).
5. Read the number of reference clock cycles (CLK_CNT0.calib_val) for calculation.

$$f_{calibrated} = \frac{cnt_val + 1}{calib_val} \bullet f_{reference} \quad (3)$$

6. Disable the module (CLK_CTRL.enable = 0).

6.4.2 Registers

6.4.2.1 SL_CLK_CTRL

- **Address**: 0xA000E400
- **Reset Value**: 0x00000000

Table 6-12 SL_CLK_CTRL Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R	0x0	Reserved bits
0	enable	RW	0x0	Enable clock calibration block. **Value:** • 0x1: Enable the block.

Bits	Field Name	RW	Reset	Description
				• 0x0: Disable the block.

6.4.2.2 SL_CLK_CNT

- **Address**: 0xA000E404
- **Reset Value**: 0x00000000

Table 6-13 SL_CLK_CNT Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R	0x0	Reserved bits
11:0	cnt_val	RW	0x0	Number of cycles of the clock being calibrated

6.4.2.3 SL_CLK_STAT

- **Address:** 0xA000E408
- **Reset Value:** 0x00000000

Table 6-14 SL_CLK_STAT Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R	0x0	Reserved bits
1	overflow	R	0x0	Fast clock counter overflow. Need to reduce slow clock cnt_val.
0	ready	R	0x0	Cal done status

6.4.2.4 SL_CLK_CNT0

- **Address:** 0xA000E40C
- **Reset Value:** 0x00000000

Table 6-15 SL_CLK_CNT0 Register

Bits	Field Name	RW	Reset	Description
31:24	RSVD	R	0x0	Reserved bits
23:0	calib_val	R	0x0	Calibrated counter value

6.4.2.5 SL_CLK_CNT1

- **Address:** 0xA000E410
- **Reset Value:** 0x00000000

Table 6-16 SL_CLK_CNT1 Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R	0x0	Reserved bits
11:0	cnt_val_rd	R	0x0	Read the current counter value of the clock being calibrated.

6.4.2.6 SL_CLK_INT_EN

- **Address:** 0xA000E414
- **Reset Value:** 0x00000003

Table 6-17 SL_CLK_INT_EN Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R	0x0	Reserved bits
1:0	val	RW	0x3	Interrupt Enable ONLY. The status in SL_CLK_STAT can be read no matter whether the interrupt is enabled.

6.4.2.7 SL_CLK_INT_CLR

- **Address:** 0xA000E418
- **Reset Value:** 0x00000000

Table 6-18 SL_CLK_INT_CLR Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R	0x0	Reserved bits
1:0	en	W	0x0	Write corresponding bit to clear.

6.4.2.8 SL_CLK_SEL

- **Address:** 0xA000E420
- **Reset Value:** 0x00000000

Table 6-19 SL_CLK_SEL Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R	0x0	Reserved bits
7:0	val	RW	0x0	Select slow clock source. Only valid for the defined source value. **Value:** • 0x0: LFXO_32K

Bits	Field Name	RW	Reset	Description
				• 0x1: LFRC_32K • 0x2: RNG_32K from RNG_OSC

6.4.2.9 HS_CLK_CTRL

- **Address:** 0xA000E500
- **Reset Value:** 0x00000000

Table 6-20 HS_CLK_CTRL Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R	0x0	Reserved bits
0	enable	RW	0x0	Enable clock calibration block. **Value:** • 0x1: Enable the block. • 0x0: Disable the block.

6.4.2.10 HS_CLK_CNT

- **Address:** 0xA000E504
- **Reset Value:** 0x00000000

Table 6-21 HS_CLK_CNT Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R	0x0	Reserved bits
11:0	cnt_val	RW	0x0	Number of cycles of the clock being calibrated

6.4.2.11 HS_CLK_STAT

- **Address:** 0xA000E508
- **Reset Value:** 0x00000000

Table 6-22 HS_CLK_STAT Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R	0x0	Reserved bits
1	overflow	R	0x0	Fast clock counter overflow. Need to reduce slow clock cnt_val.
0	ready	R	0x0	Cal done status

6.4.2.12 HS_CLK_CNT0

- **Address:** 0xA000E50C
- **Reset Value:** 0x00000000

Table 6-23 HS_CLK_CNT0 Register

Bits	Field Name	RW	Reset	Description
31:24	RSVD	R	0x0	Reserved bits
23:0	calib_val	R	0x0	Calibrated counter value

6.4.2.13 HS_CLK_CNT1

- **Address:** 0xA000E510
- **Reset Value:** 0x00000000

Table 6-24 HS_CLK_CNT1 Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R	0x0	Reserved bits
11:0	cnt_val_rd	R	0x0	Read the current counter value of the clock being calibrated.

6.4.2.14 HS_CLK_INT_EN

- **Address:** 0xA000E514
- **Reset Value:** 0x00000003

Table 6-25 HS_CLK_INT_EN Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R	0x0	Reserved bits
1:0	val	RW	0x3	Interrupt Enable ONLY. The status in SL_CLK_STAT can be read no matter whether the interrupt is enabled.

6.4.2.15 HS_CLK_INT_CLR

- **Address:** 0xA000E518
- **Reset Value:** 0x00000000

Table 6-26 HS_CLK_INT_CLR Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R	0x0	Reserved bits

Bits	Field Name	RW	Reset	Description
1:0	en	W	0x0	Write corresponding bit to clear.

6.4.2.16 HS_CLK_SEL

- **Address:** 0xA000E520
- **Reset Value:** 0x00000000

Table 6-27 HS_CLK_SEL Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R	0x0	Reserved bits
7:0	val	RW	0x0	Select slow clock source. Only valid for the defined source value. Value: • 0x0: LFXO_32K • 0x1: LFRC_32K • 0x2: XO 31.25 kHz from HFXO_32M

6.5 Registers

6.5.1 MCU_CLK_CTRL

- **Address:** 0xA000A000
- **Reset Value:** 0x42040202

Table 6-28 MCU_CLK_CTRL Register

Bits	Field Name	RW	Reset	Description
31	RSVD	R	0x0	Reserved bit
30	pmu_rng_2mhz_clk_en	RW	0x1	Enable/Disable RNG OSC 2 MHz CLK in PMU. This bit will be forcibly enabled when aon_slow_clk_sel = 1.
29	RSVD	R	0x0	Reserved bits
28	aon_slow_clk_sel	RW	0x0	Select the source of aon_slow CLK. Please allow ~10 µs for switching. **Value:** • 0x0: 250K RNG OSC CLK • 0x1: 2M RNG OSC CLK
27	RSVD	R	0x0	Reserved bit
26:24	sys_clk_sel_rd	R	0x2	Current sys_clk_sel value

Bits	Field Name	RW	Reset	Description
23:19	RSVD	R	0x0	Reserved bits
18:16	ser_clk_sel	RW	0x4	Select the clock for serial modules. (UART, I2C, SPI_M, SPI_S) **Value:** • 0x0: PLL/HF_OSC 96 MHz CLK • 0x1: PLL/HF_OSC 64 MHz CLK • 0x2: PLL/HF_OSC 32 MHz CLK • 0x3: PLL/HF_OSC 16 MHz CLK • 0x4: XO 16 MHz CLK
15:13	RSVD	R	0x0	Reserved bits
12	wkup_sys_clk_sel_en	RW	0x0	Apply the wkup_sel to sys_clk_sel on wake-up.
11	RSVD	R	0x0	Reserved bit
10:8	wkup_sys_clk_sel	RW	0x2	System clock selection on wake-up
7:3	RSVD	R	0x0	Reserved bits
2:0	sys_clk_sel	RW	0x2	Select the system clock for MCU subsystem. **Value:** • 0x0: PLL/HF_OSC 96 MHz CLK • 0x1: PLL/HF_OSC 64 MHz CLK • 0x2: XO 16 MHz CLK • 0x3: PLL/HF_OSC 48 MHz CLK • 0x4: PLL/HF_OSC 24 MHz CLK • 0x5: PLL/HF_OSC 16 MHz CLK • 0x6: PLL/HF_OSC 32 MHz CLK

6.5.2 AON_CLK

- **Address:** 0xA000A018
- **Reset Value:** 0x00000000

Table 6-29 AON_CLK Register

Bits	Field Name	RW	Reset	Description
31:9	RSVD	R	0x0	Reserved bits
8	wkup_aon_fast_clk_sel	RW	0x0	Select the source of the 192 MHz clock on wakeup. **Value:** • 0x0: CPLL_192M • 0x1: HFRC_192M

Bits	Field Name	RW	Reset	Description
7:5	RSVD	R	0x0	Reserved bits
4	wkup_aon_fast_clk_sel_en	RW	0x0	Apply wkup_aon_fast_clk_sel value to reg_aon_fast_clk_sel on wakeup.
3:1	RSVD	R	0x0	Reserved bits
0	reg_aon_fast_clk_sel	RW	0x0	Select the source of the 192 MHz clock. **Value:** • 0x0: CPLL_192M • 0x1: HFRC_192M

6.5.3 COMM_CTRL

- **Address:** 0xA000A280
- **Reset Value:** 0x00020000

Table 6-30 COMM_CTRL Register

Bits	Field Name	RW	Reset	Description
31:18	RSVD	R	0x0	Reserved bits
17:16	comm_timer_clk_sel	RW	0x2	Select the Comm timer clock. **Value:** • 0x0: LFXO_32K • 0x1: LFRC_32K • 0x2: RNG_OSC • 0x3: RNG_OSC
15:10	RSVD	R	0x0	Reserved bits
9:4	comm_timer_deepslcntl_wr	RW	0x0	Comm timer register
3	RSVD	R	0x0	Reserved bit
2	comm_timer_deepslcntl_rd	R	0x0	Comm timer register
1:0	RSVD	R	0x0	Reserved bits

6.5.4 ANA_SLP_CFG

- **Address:** 0xA000A420
- **Reset Value:** 0x00030017

Table 6-31 ANA_SLP_CFG Register

Bits	Field Name	RW	Reset	Description
31:18	RSVD	R	0x0	Reserved bits
17	slp_trn_off_pll	RW	0x1	PLL status in deep sleep mode **Value:** • 0x1: Turn off. • 0x0: Keep current status.
16	slp_trn_off_xo	RW	0x1	XO status in deep sleep mode **Value:** • 0x1: Turn off. • 0x0: Keep current status.
15:5	RSVD	R	0x0	Reserved bits
4	slp_trn_off_fast_ldo	RW	0x1	FAST_LDO status in deep sleep mode **Value:** • 0x1: Turn off. • 0x0: Keep current status.
3	slp_trn_off_io_ldo	RW	0x0	IO_LDO status in deep sleep mode **Value:** • 0x1: Turn off. • 0x0: Keep current status.
2	slp_trn_off_hf_osc	RW	0x1	HF_OSC status in deep sleep mode **Value:** • 0x1: Turn off. • 0x0: Keep current status.
1	slp_trn_off_dig_ldo	RW	0x1	Dig_LDO status in deep sleep mode **Value:** • 0x1: Turn off. • 0x0: Keep current status.
0	slp_trn_off_dcdc	RW	0x1	DCDC status in deep sleep mode **Value:** • 0x1: Turn off. • 0x0: Keep current status.

6.5.5 XO_PLL_SET

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

- **Address:** 0xA000A440
- **Reset Value:** 0x00000000

Table 6-32 XO_PLL_SET Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R	0x0	Reserved bits
2	hf_osc_set	W	0x0	Write '1' to enable HFRC_192M. This operation will turn DCDC on if it is not.
1	xo_req_set	W	0x0	Write '1' to enable HFXO_32M. This operation will turn DCDC on if it is not.
0	pll_req_set	W	0x0	Write '1' to enable CPLL_192M. This operation will turn DCDC on if it is not.

6.5.6 XO_PLL_CLR

- **Address:** 0xA000A444
- **Reset Value:** 0x00000000

Table 6-33 XO_PLL_CLR Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R	0x0	Reserved bits
2	hf_osc_clr	W	0x0	Write '1' to disable HFRC_192M.
1	xo_req_clr	W	0x0	Write '1' to disable HFXO_32M.
0	pll_req_clr	W	0x0	Write '1' to disable CPLL_192M.

6.5.7 XO_PLL_STAT

- **Address:** 0xA000A448
- **Reset Value:** 0x00000000

Table 6-34 XO_PLL_STAT Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R	0x0	Reserved bits
2	hf_osc_avl	R	0x0	Read HFRC_192M status.
1	xo_req_avl	R	0x0	Read HFXO_32M status.
0	pll_req_avl	R	0x0	Read CPLL_192M status.

6.5.8 MCU_SUBSYS_CG_CTRL_1

- **Address:** 0xA000EE84
- **Reset Value:** 0x0000FFEF

Table 6-35 MCU_SUBSYS_CG_CTRL_1 Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R	0x0	Reserved bits
15:13	reserved	RW	0x7	Reserved bits
12	mcu_sub_usb_hclk	RW	0x1	HCLK for USB **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
11	reserved	RW	0x1	Reserved bit
10	mcu_sub_serial_hclk	RW	0x1	HCLK for serial blocks, including I2C, UART, QSPI, SPI (M&S), and I2S **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
9	mcu_sub_apb_sub_hclk	RW	0x1	HCLK for APB subsystem: watchdog, timer, etc. **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
8	mcu_sub_ble_brg_hclk	RW	0x1	HCLK for BLE MCU bridge **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
7	reserved	RW	0x1	Reserved bit
6	mcu_sub_gpio_hclk	RW	0x1	HCLK for GPIOs **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
5	mcu_sub_snsadc_hcl	RW	0x1	HCLK for sense ADC **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
4	mcu_sub_rom_hclk	RW	0x0	HCLK for ROM

Bits	Field Name	RW	Reset	Description
				Value: • 0x1: Force clock OFF. • 0x0: Force clock ON.
3	reserved	RW	0x1	Reserved bit
2	mcu_sub_htb_hclk	RW	0x1	HCLK for hopping table **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
1	mcu_sub_sim_hclk	RW	0x1	HCLK for SIM card interface **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
0	mcu_sub_secu_hclk	RW	0x1	HCLK for all security blocks, including: AES, PKC, Present, and HMAC **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.

6.5.9 MCU_SUBSYS_CG_CTRL_2

- **Address:** 0xA000EE88
- **Reset Value:** 0x00020007

Table 6-36 MCU_SUBSYS_CG_CTRL_2 Register

Bits	Field Name	RW	Reset	Description
31:19	RSVD	R	0x0	Reserved bits
18	sram_hclk	RW	0x7	HCLK for SRAMs **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
17	cache_top_hclk	RW	0x1	HCLK for cache top, including XF_XQSPI and XQSPI cache **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
16	mcu_reg_hclk	RW	0x0	HCLK for mcu_aux_reg and always_on_reg

Bits	Field Name	RW	Reset	Description
				Value: • 0x1: Force clock OFF. • 0x0: Force clock ON.
15:3	RSVD	R	0x0	Reserved bits
2:0	reserved	RW	0x7	Reserved bits

6.5.10 MCU_PERIPH_CLK_OFF

- **Address:** 0xA000EE8C
- **Reset Value:** 0xFF7FCFFF

Table 6-37 MCU_PERIPH_CLK_OFF Register

Bits	Field Name	RW	Reset	Description
31	clk_ctrl_psram_ctrl_pclk_force_off	RW	0x1	PSRAM CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
30	clk_ctrl_vttbl_force_off	RW	0x1	VTTBL CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
29	clk_ctrl_pwm_1_force_off	RW	0x1	PWM 1 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
28	clk_ctrl_pwm_0_force_off	RW	0x1	PWM 0 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
27	reg_xf_xqspi_div4_pclk_force_off	RW	0x1	div4 CLK for XF QSPI **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
26	reserved	RW	0x1	Reserved bit
25	reg_secu_div4_pclk_force_off	RW	0x1	div4 CLK for security blocks

Bits	Field Name	RW	Reset	Description
				Value: • 0x1: Force clock OFF. • 0x0: Force clock ON.
24	reserved	RW	0x1	Reserved bit
23	RSVD	R	0x0	Reserved bit
22	clk_ctrl_pdm_clk_force_off	RW	0x1	PDM CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
21	clk_ctrl_dspi_pclk_force_off	RW	0x1	DSPI CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
20	clk_ctrl_i2s_s_pclk_force_off	RW	0x1	i2s_s CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
19	clk_ctrl_i2s_pclk_force_off	RW	0x1	i2s CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
18	clk_ctrl_spi_s_clk_force_off	RW	0x1	spi_s CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
17	clk_ctrl_spi_m_clk_force_off	RW	0x1	spi_m CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
16	clk_ctrl_qspi_2_pclk_force_off	RW	0x1	OSPI CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
15	clk_ctrl_qspi_1_pclk_force_off	RW	0x1	qspi_1 CLK

Bits	Field Name	RW	Reset	Description
				Value: • 0x1: Force clock OFF. • 0x0: Force clock ON.
14	clk_ctrl_qspi_0_pclk_force_off	RW	0x1	qspi_0 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
13:12	RSVD	R	0x0	Reserved bits
11	clk_ctrl_i2c_5_clk_force_off	RW	0x1	i2c_5 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
10	clk_ctrl_i2c_4_clk_force_off	RW	0x1	i2c_4 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
9	clk_ctrl_i2c_3_clk_force_off	RW	0x1	i2c_3 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
8	clk_ctrl_i2c_2_clk_force_off	RW	0x1	i2c_2 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
7	clk_ctrl_i2c_1_clk_force_off	RW	0x1	i2c_1 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
6	clk_ctrl_i2c_0_clk_force_off	RW	0x1	i2c_0 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
5	clk_ctrl_uart_5_force_off	RW	0x1	uart_5 CLK **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x1: Force clock OFF. • 0x0: Force clock ON.
4	clk_ctrl_uart_4_force_off	RW	0x1	uart_4 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
3	clk_ctrl_uart_3_force_off	RW	0x1	uart_3 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
2	clk_ctrl_uart_2_force_off	RW	0x1	uart_2 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
1	clk_ctrl_uart_1_force_off	RW	0x1	uart_1 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
0	clk_ctrl_uart_0_force_off	RW	0x1	uart_0 CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.

6.5.11 MCU_PERIPH_CG_LP_EN

- **Address:** 0xA000EE90
- **Reset Value:** 0x00000F3D

Table 6-38 MCU_PERIPH_CG_LP_EN Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R	0x0	Reserved bits
11	reg_apb2apb_async_brg_lp_en	RW	0x1	Enable APB2APB async bridge low-power feature.
10	reg_ahb2apb_brg_lp_en	RW	0x1	Enable AHB2APB sync bridge low-power feature.
9	reg_qspi_m_lp_sclk_en	RW	0x1	Enable QSPI master SCLK low-power feature.
8	reg_ahb_bus_lp_en	RW	0x1	Enable AHB bus low-power feature.

Bits	Field Name	RW	Reset	Description
7:6	RSVD	R	0x0	Reserved bits
5	reg_i2c_lp_sclk_en	RW	0x1	Enable I2C SCLK low-power feature.
4	reg_spi_s_lp_sclk_en	RW	0x1	Enable SPI slave SCLK low-power feature.
3	reg_spi_m_lp_sclk_en	RW	0x1	Enable SPI master low-power feature.
2	reg_i2s_lp_en	RW	0x1	Enable I2S master low-power feature.
1	reg_uart_lp_pclk_en	RW	0x0	Enable UART PCLK low-power feature.
0	reg_uart_lp_sclk_en	RW	0x1	Enable UART SCLK low-power feature.

6.5.12 MCU_SUBSYS_SECU_CLK_CTRL

- **Address:** 0xA000EE98
- **Reset Value:** 0x00003FFF

Table 6-39 MCU_SUBSYS_SECU_CLK_CTRL Register

Bits	Field Name	RW	Reset	Description
31:14	RSVD	R	0x0	Reserved bits
13	reserved	RW	0x1	Reserved bit
12	cfg_efuse_clk_force_off_en	RW	0x1	eFuse CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
11	reserved	RW	0x1	Reserved bit
10	cfg_rng_clk_force_off_en	RW	0x1	RNG CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
9	reserved	RW	0x1	Reserved bit
8	cfg_ramkey_clk_force_off_en	RW	0x1	KeyRAM CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
7	reserved	RW	0x1	Reserved bit
6	cfg_present_clk_force_off_en	RW	0x1	Present CLK **Value:** • 0x1: Force clock OFF.

Bits	Field Name	RW	Reset	Description
				• 0x0: Force clock ON.
5	reserved	RW	0x1	Reserved bit
4	cfg_pkc_clk_force_off_en	RW	0x1	PKC CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
3	reserved	RW	0x1	Reserved bit
2	cfg_hmac_clk_force_off_en	RW	0x1	HMAC CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
1	reserved	RW	0x1	Reserved bit
0	cfg_aes_clk_force_off_en	RW	0x1	AES CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.

6.5.13 MCU_SUBSYS_MISC_CLK_CTRL

- **Address:** 0xA000EE9C
- **Reset Value:** 0x0000003F

Table 6-40 MCU_SUBSYS_MISC_CLK_CTRL Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R	0x0	Reserved bits
5	clk_ctrl_dma_2_clk_off	RW	0x1	DMA 2 HCLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
4	clk_ctrl_dma_1_clk_off	RW	0x1	DMA 1 HCLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
3	clk_ctrl_dma_0_clk_off	RW	0x1	DMA 1 HCLK **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x1: Force clock OFF. • 0x0: Force clock ON.
2	clk_ctrl_xf_sck_clk_off	RW	0x1	XQSPI SCK CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
1	clk_ctrl_gpu_clk_off	RW	0x1	GPU clock **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.
0	clk_ctrl_dc_clk_off	RW	0x1	Display controller CLK **Value:** • 0x1: Force clock OFF. • 0x0: Force clock ON.

6.5.14 MCC_PLL_HFOSC_CLK_EN

- **Address:** 0xA000EEC0
- **Reset Value:** 0x00000001

Table 6-41 MCC_PLL_HFOSC_CLK_EN Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R	0x0	Reserved bits
0	mcc_pll_hfosc_clk_en	RW	0x1	This clock source can be cut off if only XO clock is used.

6.5.15 MCC_CLK_FORCE_ON

- **Address:** 0xA000EEC4
- **Reset Value:** 0x00000000

Table 6-42 MCC_CLK_FORCE_ON Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R	0x0	Reserved bits
7	reg_xo_16_force_on	RW	0x0	**Value:** • 0x1: Force XO 16 MHz CLK on. • 0x0: Automatically controlled.

Bits	Field Name	RW	Reset	Description
6	reg_xo_32_force_on	RW	0x0	**Value:** • 0x1: Force XO 32 MHz CLK on. • 0x0: Automatically controlled.
5	reg_pll_16_force_on	RW	0x0	**Value:** • 0x1: Force cpll_hf_osc 16 MHz CLK on. • 0x0: Automatically controlled.
4	reg_pll_24_force_on	RW	0x0	**Value:** • 0x1: Force cpll_hf_osc 24 MHz CLK on. • 0x0: Automatically controlled.
3	reg_pll_32_force_on	RW	0x0	**Value:** • 0x1: Force cpll_hf_osc 32 MHz CLK on. • 0x0: Automatically controlled.
2	reg_pll_48_force_on	RW	0x0	**Value:** • 0x1: Force cpll_hf_osc 48 MHz CLK on. • 0x0: Automatically controlled.
1	reg_pll_64_force_on	RW	0x0	**Value:** • 0x1: Force cpll_hf_osc 64 MHz CLK on. • 0x0: Automatically controlled.
0	reg_pll_96_force_on	RW	0x0	**Value:** • 0x1: Force cpll_hf_osc 96 MHz CLK on. • 0x0: Automatically controlled.

7 Always-on Domain (AON)

7.1 Always-on Watchdog (AON WDT)

7.1.1 Introduction

The Always-on Watchdog Timer (AON_WDT) is a 32-bit hardware timer that generates a system-level reset or interrupts if the firmware fails to set it in time. Since AON_WDT is in AON domain, the timer can keep running when the MCU is off.

7.1.2 Main Features

- The CLK register is a read/write register when the AON_WDT is disabled and is read only when the AON_WDT is enabled. Writing to the CLK register when the AON_WDT is enabled has no effect.
- 32-bit down counter running with a low-power clock
- Generates an interrupt to warn the MCU when the 32-bit down counter reaches alarm values
- 16-bit alarm register, supporting reading and writing
- Flexible clock source selections from LFRC_32K or LFXO_32K
- An independent interrupt entry
- Supports write access protection

7.1.3 Functional Description

7.1.3.1 Block Diagram

The following figure shows the functional blocks of the Always-on watchdog module.

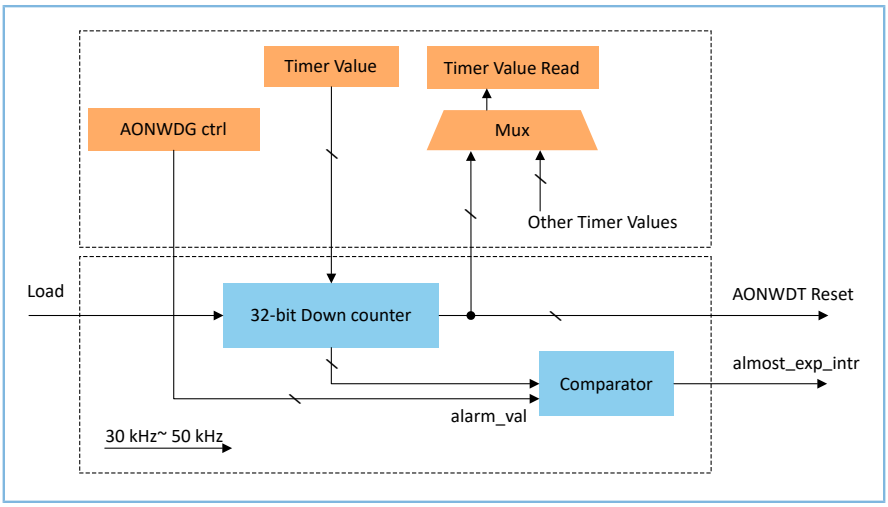

Figure 7-1 Always-on Watchdog Block Diagram

The watchdog starts running when `AON_WDT_EN = 1` is synchronized to AON_WDT clock domain, which takes 1 – 3 RNG (Random Number Generators) clock cycles (20 – 100 µs).

If no reload operation is performed, the down counter default value is **0xFFFFFFFF**. Otherwise, the counter will be reloaded with the value of timer_value.

7.1.3.2 Clock

The AON_WDT works in RNG clock (30 kHz – 50 kHz), which is much slower than MCU system clock (16 MHz – 64 MHz). Therefore, synchronization time is needed to propagate the MCU operations. For example, in reload operation, MCU needs to check whether the **VAL_RD** value has changed as expected.

7.1.3.3 Enable/Disable and Timer Reload

- To enable the block, set `AON_WDT_EN = 1` and wait until **AON_WDT_RUNNING** being high.
- To disable the block, set `AON_WDT_EN = 0` and wait until **AON_WDT_RUNNING** being low.
- To reload down counter, program the desired value to timer_value, then set write-only register `AON_WDT_RELOAD = 1`. MCU can check read-only register **VAL_RD** to confirm.

7.1.3.4 Alarm

The AON_WDT provides an almost-expired alarm to warn MCU before system reset. By setting **ALARM_VAL= N(non-zero)**, the AON_WDT generates an interrupt N RNG clocks before the system reset.

7.1.4 Registers

7.1.4.1 CFG_0

- **Name:** CFG_0 Register
- **Description:** This register sets AON_WDT configurations.
- **Base Address:** 0x4000A700
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 7-1 CFG_0 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R	0x0	Reserved bits
24	CFG	W	0x0	Writing 1 to this register triggers the watchdog timer configuration.
23:4	RSVD	R	0x0	Reserved bits
3	ALARM_SET	RW	0x0	Sets the timer alarm value.
2	ALARM_EN	RW	0x0	Enables AON WDT alarm.
1	RELOAD	W	0x0	Enables AON WDT timer reload (feed).
0	EN	RW	0x0	Enables AON WDT.

7.1.4.2 LOCK

- **Name:** LOCK Register
- **Description:** This register can prevent other registers of AON_WDT from being overwritten.
- **Base Address:** 0x4000A700
- **Offset:** 0x4
- **Reset Value:** 0x00000000

Table 7-2 LOCK Register

Bits	Field Name	RW	Reset	Description
31:1	LOCK_VALUE	W	–	Enable write access to all other registers by writing 0x15CC5A51. Disable write access by writing any other value to the LOCK register.
0	STATUS	R	0x0	• 0: Unlocked. Write to other watchdog registers is enabled. • 1: Locked. Write to other watchdog registers is disabled.

7.1.4.3 STAT

- **Name:** STAT Register

- **Description:** This register can be used to get status of AON_WDT.
- **Base Address:** 0x4000A700
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 7-3 STAT Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R	0x0	Reserved bits
1	BUSY	R	0x0	Watchdog timer is busy. The application layer can perform other operations until this bit is set is 0.
0	STAT	R	0x0	AON WDT is running.

7.1.4.4 CLK

- **Name:** CLK Register
- **Description:** This register is used to select an AON_WDT clock source.
- **Base Address:** 0x4000A700
- **Offset:** 0x3C
- **Reset Value:** 0x00000000

Table 7-4 CLK Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R	0x0	Reserved bits
1:0	CLK_SEL	RW	0x0	• 0x02: LFRC_32K (LFRC 32 kHz Clock) • 0x03: LFXO_32K (LF_XO 32 kHz Clock)

7.1.4.5 TIMER_WR

- **Name:** TIMER_WR Register
- **Description:** This register is used to configure the initial AON_WDT value.
- **Base Address:** 0x4000A700
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 7-5 TIMER_WR Register

Bits	Field Name	RW	Reset	Description
31:0	TIMER_SET	RW	0x0	**Write**:

Bits	Field Name	RW	Reset	Description
				Write the initial AON WDT value to the register. **Read**: • Read the initial AON WDT value that users have configured. • Read TIMER_RD to get the current internal value.

7.1.4.6 ALARM_WR

- **Name:** ALARM_WR Register
- **Description:** This register is used to configure the AON_WDT alarm value.
- **Base Address:** 0x4000A700
- **Offset:** 0x44
- **Reset Value:** 0x64

Table 7-6 ALARM_WR Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R	0x0	Reserved bits
15:0	ALARM_SET	RW	0x64	**Write**: Write an AON WDT alarm value to the register. **Read**: • Read the AON WDT alarm value that users have configured. • Read ALARM_RD to get the current internal alarm value of AON WDT.

7.1.4.7 TIMER_RD

- **Name:** TIMER_RD Register
- **Description:** This register is used to get the current value of AON_WDT.
- **Base Address:** 0x4000A700
- **Offset:** 0x4C
- **Reset Value:** 0x00000000

Table 7-7 TIMER_RD Register

Bits	Field Name	RW	Reset	Description
31:0	TIMER_RD	R	0x0	Current internal value of AON WDT

7.1.4.8 ALARM_RD

- **Name:** ALARM_RD Register
- **Description:** This register reads the current alarm value of AON_WDT.
- **Base Address:** 0x4000A700
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 7-8 ALARM_RD Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R	0x0	Reserved bits
15:0	ALARM_RD	R	0x0	Current internal alarm value of the AON WDT

7.2 Real-Time Counter (RTC/Calendar)

The Real-time Counter (RTC) modules provide a generic, low-power timer running on a low-frequency clock source. Since the RTC modules are in AON domain, the timer can keep running when the MCU is off. The modules can also wake up the system from Deep Sleep.

There are two identical RTC modules: one for calendar and the other for RTOS.

The RTC tick event interrupt module enables low power "tick-less" RTOS implementation as it optionally provides a regular interrupt source for an RTOS without the need to use the ARM® SysTick feature. Using the RTC tick event rather than the SysTick allows the CPU to be powered down while still keeping RTOS scheduling active.

7.2.1 Main Features

- One ordinary compare interrupt
- One tick event, supporting one-time counting and auto loading mode
- 32-bit up-counter register, supporting reading and writing
- 32-bit compare register, supporting reading and writing
- 32-bit down-tick register, supporting reading and writing
- One clock divider, supporting NO_DIV, DIV_2, DIV_4, DIV_8, DIV_16, DIV_32, DIV_64, and DIV_128
- Flexible clock source selections from LFRC_32K or LFXO_32K
- An independent interrupt entry

7.2.2 Functional Description

- The software has to enable RTC PMU LPD (RTC_EN) before using the RTC.
- The CLK_DIV/CLK_SEL register is a read/write register when the RTC is disabled and is read only when the RTC is enabled. Writing to the CLK_DIV/CLK_SEL register when the RTC is started has no effect.

- The OVERFLOW, COMPARE, and tick event interrupts are enabled by default.
- When RTC is enabled, the TIMER_RD register starts to count. When users set the values of the TIMER_SET register and TIME_SET bit to 1, the calendar module synchronizes the count value to the TIMER_RD register and starts timing.
- When RTC is disabled, all register values must be cleared to the default values.
- When the value of the TIMER_RD register is equal to that of the CMP_RD register, the ordinary compare interrupt is triggered.
- RTC tick event interrupt register is a value-decreasing register. When the auto reloading mode is set (TICK_TYPE), the value of the counter (TICK_SET) is automatically reloaded (TICK_RD) when the interrupt is triggered.
- When the count value of the TIMER_RD register overflows, the overflow interrupt is triggered, and the TIMER_RD register continues counting from 0.

7.2.3 Registers

7.2.3.1 CTRL

- **Name:** CTRL Register
- **Description:** This register sets RTC configurations.
- **Base Address:** 0x4000A600 + x*0x80
- **Offset:** 0x00
- **Reset Value:** 0x00000000

Table 7-9 CTRL Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R	0x0	Reserved bits
24	CFG	RW	0x0	Writing 1 to this register triggers the calendar timer configuration.
23:11	RSVD	R	0x	Reserved bits
10	TICK_TYPE	RW	0x0	Sets the RTC tick event mode. • 0x0: One-time • 0x1: Auto reloading
9	TICK_SET	RW	0x0	Sets the RTC tick event timer value.
8	TICK_EN	RW	0x0	Enables RTC tick event.
7:5	RSVD	R	0x0	Reserved bits
4	WRAP_CLR	RW	0x0	Clears wrap counter.
3	CMP_SET	RW	0x0	Sets the RTC COMPARE timer value.
2	COMPARE_EN	RW	0x0	Enables RTC COMPARE timer.

Bits	Field Name	RW	Reset	Description
1	TIME_SET	RW	0x0	Sets the RTC initial timer value.
0	EN	RW	0x0	Enables RTC timer.

7.2.3.2 CLK_DIV

- **Name:** CLK_DIV Register
- **Description:** This register is used to set the frequency division coefficient
- **Base Address:** 0x4000A600 + x*0x80
- **Offset:** 0x4
- **Reset Value:** 0x00000000

Table 7-10 CLK_DIV Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R	0x0	Reserved bits
2:0	CLK_DIV	RW	0x0	• 0x0: NO_DIV. Not divided. Counting at SLP_CLK • 0x1: DIV_2. Counting at 1/2 of SLP_CLK • 0x2: DIV_4. Counting at 1/4 of SLP_CLK • 0x3: DIV_8. Counting at 1/8 of SLP_CLK • 0x4: DIV_16. Counting at 1/16 of SLP_CLK • 0x5: DIV_32. Counting at 1/32 of SLP_CLK • 0x6: DIV_64. Counting at 1/64 of SLP_CLK • 0x7: DIV_128. Counting at 1/128 of SLP_CLK

7.2.3.3 IRQ_EN

- **Name:** IRQ_EN Register
- **Description:** This register is used to enable RTC interrupts.
- **Base Address:** 0x4000A600 + x*0x80
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 7-11 IRQ_EN Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R	0x0	Reserved bits
2	TICK_EVT	RW	0x1	RTC tick event interrupt. Write 1 to enable the interrupt.

Bits	Field Name	RW	Reset	Description
1	OVERFLOW	RW	0x1	RTC overflow interrupt. Write 1 to enable the interrupt.
0	COMPARE	RW	0x1	RTC COMPARE interrupt. Write 1 to enable the interrupt.

7.2.3.4 IRQ_STAT

- **Name:** IRQ_STAT Register
- **Description:** This register is used to get interrupt status of the RTC.
- **Base Address:** 0x4000A600 + x*0x80
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 7-12 IRQ_STAT Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R	0x0	Reserved bits
2	TICK_EVT	RW	0x0	RTC tick event interrupt. Write 1 to clear the interrupt.
1	OVERFLOW	RW	0x0	RTC overflow interrupt. Write 1 to clear the interrupt.
0	COMPARE	RW	0x0	RTC COMPARE interrupt. Write 1 to clear the interrupt.

7.2.3.5 STAT

- **Name:** STAT Register
- **Description:** This register is used to get the RTC status.
- **Base Address:** 0x4000A600 + x*0x80
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 7-13 STAT Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R	0x0	Reserved bits
11:8	WRAP_CNT	R	0x0	Calendar timer wrap-around counter
7:5	RSVD	R		Reserved bits
4	BUSY	R	0x0	Calendar timer is busy. The application layer can perform other operations until this bit is set is 0.
3:1	RSVD	R		Reserved bits
0	STAT	R	0x0	RTC timer is running.

7.2.3.6 CLK_SEL

- **Name:** CLK_SEL Register
- **Description:** This register is used to select a clock source.
- **Base Address:** 0x4000A600 + x*0x80
- **Offset:** 0x3C
- **Reset Value:** 0x00000000

Table 7-14 CLK_SEL Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R	0x0	Reserved bits
1:0	CLK_SEL	RW	0x0	• 0x02: LFRC_32K (LFRC 32 kHz Clock) • 0x03: LFXO_32K (LF_XO 32 kHz Clock)

7.2.3.7 CMP_SET

- **Name:** CMP_SET Register
- **Description:** This register is used to set the COMPARE timer value.
- **Base Address:** 0x4000A600 + x*0x80
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 7-15 CMP_SET Register

Bits	Field Name	RW	Reset	Description
31:0	CMP_SET	RW	0x0	• Write: RTC COMPARE timer value to configure • Read: The value you just set. Read CMP_RD to get the current internal value.

7.2.3.8 TIMER_SET

- **Name:** TIMER_SET Register
- **Description:** This register is used to set the RTC timer value.
- **Base Address:** 0x4000A600 + x*0x80
- **Offset:** 0x44
- **Reset Value:** 0x00000000

Table 7-16 TIMER_SET Register

Bits	Field Name	RW	Reset	Description
31:0	TIMER_SET	RW	0x0	• Write: RTC timer value to configure • Read: The value you just set. Read TIMER_RD to get the current internal value.

7.2.3.9 TICK_SET

- **Name**: TICK_SET Register
- **Description**: This register is used to set the tick event timer value
- **Base Address**: 0x4000A600 + x*0x80
- **Offset**: 0x48
- **Reset Value**: 0x00000000

Table 7-17 TICK_SET Register

Bits	Field Name	RW	Reset	Description
31:0	TICK_SET	RW	0x0	• Write: RTC tick event timer value to configure • Read: The value you just set. Read TICK_RD to get the current internal value.

7.2.3.10 TIMER_RD

- **Name:** TIMER_RD Register
- **Description:** This register is used to get the internal value of the RTC timer.
- **Base Address:** 0x4000A600 + x*0x80
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 7-18 TIMER_RD Register

Bits	Field Name	RW	Reset	Description
31:0	TIMER_RD	R	0x0	Current internal value of the RTC timer

7.2.3.11 CMP_RD

- **Name:** CMP_RD Register
- **Description:** This register is used to get the value of the RTC COMPARE timer.
- **Base Address:** 0x4000A600 + x*0x80
- **Offset:** 0x54

- **Reset Value:** 0x00000000

Table 7-19 CMP_RD Register

Bits	Field Name	RW	Reset	Description
31:0	CMP_RD	R	0x0	Current internal value of the RTC COMPARE timer

7.2.3.12 TICK_RD

- **Name:** TICK_RD Register
- **Description:** This register used to get the value of the RTC tick event timer.
- **Base Address:** 0x4000A600 + x*0x80
- **Offset:** 0x58
- **Reset Value:** 0x00000000

Table 7-20 TICK_RD Register

Bits	Field Name	RW	Reset	Description
31:0	TICK_RD	RW	0x0	Current internal value of the RTC tick event timer.

7.3 Sleep Timer (SLP TIMER)

7.3.1 Introduction

The Sleep Timer is a 32-bit hardware timer that can wake the SoC up from the Deep Sleep state. It can also be used while the SoC is awake.

7.3.2 Main Features

- 32-bit down counter running with two different clock sources: LFXO_32K clock (32.768 kHz), and LFRC_32K clock (about 32 kHz).
- Three operation modes: Basic Sleep Timer, Single-load Sleep Timer, Auto-reload Sleep Timer.

7.3.3 Functional Description

The Sleep Timer supports three operation modes.

- Mode 0 – Basic Sleep Timer: This is the original mode of the Sleep Timer. It only counts down when the SoC is in Deep Sleep state and is not activated when the SoC is awake.
- Mode 1 – Single-load Sleep Timer: In this mode, the timer starts counting as soon as it is put in this mode and it continues to count down even when the SoC is awake. The timer stops when the counter hits zero. The time-out signal is saved as a timeout event.

- Mode 2 – Auto-reload Sleep Timer: Similar to Single-load Sleep Timer mode, the timer does not stop counting down even when the SoC is awake. However, the timer can automatically reload to a value that is the same as that before automatic reload until it is disabled by the MCU.

7.3.4 Registers

7.3.4.1 CFG_0

- **Base Address:** 0x4000A500
- **Offset:** 0x00
- **Reset Value:** 0x00000000

Table 7-21 CFG_0 Register

Bits	Field Name	RW	Reset	Description
24	SLP_TIME_CFG	W	N/A	**Write**: • Writing 1 to this register triggers the Sleep Timer configuration. • Writing to this bit during sleep timer configuration will generate a configuration failure interrupt. **Read**: • The read value is always 0.
3	SLP_TIMER_CNT_SLP_ONLY	RW	0x0	**Value**: • 0x1: Count during sleep only. • 0x0: Count in any condition.
2	SLP_TIMER_SINGLE_EN	RW	0x0	**Value**: • 0x1: Single-load mode • 0x0: Auto reload mode
1	SLP_TIMER_SET	RW	0x0	Sets the timer value; write 1 to make timer value effective.
0	SLP_TIMER_EN	RW	0x0	Enables Sleep Timer. **Value**: • 0x1: Enable Sleep Timer. • 0x0: Disable Sleep Timer.

7.3.4.2 STS

- **Base Address:** 0x4000A500
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 7-22 STS Register

Bits	Field Name	RW	Reset	Description
1	SLP_TIMER_BUSY	R	N/A	Sleep Timer is running.
0	SLP_TIMER_STATUS	R	N/A	Sleep Timer is busy. Do not configure this bit if the signal is 1.

7.3.4.3 CLK

- **Base Address:** 0x4000A500
- **Offset:** 0x3C
- **Reset Value:** 0x00000000

Table 7-23 CLK Register

Bits	Field Name	RW	Reset	Description
1:0	SLP_TIMER_CLK_SEL	RW	0x0	Selects Sleep Timer clock. **Value**: • 0x02: LFRC_32K (LFRC 32 kHz Clock) • 0x03: LFXO_32K (LF_XO 32 kHz Clock)

7.3.4.4 VAL_W

- **Base Address:** 0x4000A500
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 7-24 VAL_W Register

Bits	Field Name	RW	Reset	Description
31:0	SLP_TIMER_VAL_SET	RW	0x0	**Write**: • Write the Sleep Timer value to the VAL_W register. **Read**: • The value you just set. Read SLP_TIMER_VAL_RD in the VAL_R register for the current interval value.

7.3.4.5 VAL_R

- **Base Address:** 0x4000A500
- **Offset:** 0x48
- **Reset Value:** 0x00000000

Table 7-25 VAL_R

Bits	Field Name	RW	Reset	Description
31:0	SLP_TIMER_VAL_RD	R	N/A	Read internal value of the Sleep Timer.

8 Peripherals

8.1 Pin Mux

8.1.1 Introduction

GR5526 series has a configurable pin multiplexing module (Pin Mux) which can bring different peripherals on different GPIOs.

8.1.2 Functional Description

The pin multiplexing choices for all pads are shown in:

- Table 8-1 and Table 8-2 (GPIO_0–GPIO_15)
- Table 8-3 and Table 8-4 (GPIO_16–GPIO_33)
- Table 8-5 (AON_GPIO_0–AON_GPIO_7)
- Table 8-6 (MSIO_0–MSIO_7)

There are eight mux choices (from MUX_0 to MUX_7) in the pin mux tables as follows.

Table 8-1 Pin multiplexing for GPIO_0–GPIO_7

IO Name	MUX_0	MUX_1	MUX_2	MUX_3	MUX_4	MUX_5	MUX_6	MUX_7
GPIO_0	SWD_CLK	I2C1_SCL	UART0_TX	UART1_CTS	PWM1_C	PWM0_C	PDM_CLKO	GPIO_A0
GPIO_1	SWD_IO	I2C1_SDA	UART0_RX	UART1_RTS	PWM1_B	PWM0_B	PDM_DI	GPIO_A1
GPIO_2	I2C0_SCL	PDM_CLKO	UART0_CTS	UART1_TX	PWM1_A	PWM0_A	iso_sync1_p	GPIO_A2
GPIO_3	I2C0_SDA	PDM_DI	UART0_RTS	UART1_RX	FERP_GPIO_Trig	SWV	iso_sync0_p	GPIO_A3
GPIO_4	I2C4_SCL	SPI_M_CLK	PWM0_A	UART3_TX	PDM_CLKO	df_ant_switch_sel_0	UART2_CTS	GPIO_A4
GPIO_5	I2C4_SDA	SPI_M_MOSI	PWM0_B	UART3_RX	PDM_DI	df_ant_switch_sel_1	UART2_RTS	GPIO_A5
GPIO_6	I2C3_SCL	SPI_M_MISO	PWM0_C	I2S_WS	iso_sync1_p	SPI_S_CLK	SIM_PRESENCE	GPIO_A6
GPIO_7	I2C3_SDA	SPI_M_CS0_N	PWM1_A	I2S_TX_SDO	iso_sync0_p	SPI_S_MISO	SIM_RST_N	GPIO_A7

Table 8-2 Pin multiplexing for GPIO_8–GPIO_15

IO Name	MUX_0	MUX_1	MUX_2	MUX_3	MUX_4	MUX_5	MUX_6	MUX_7
GPIO_8	I2C5_SCL	UART2_TX	PWM1_B	I2S_RX_SDI	UART3_CTS	SPI_S_MOSI	SIM_IO	GPIO_A8
GPIO_9	I2C5_SDA	UART2_RX	PWM1_C	I2S_SCLK	UART3_RTS	SPI_S_CS_N	SIM_CLK	GPIO_A9
GPIO_10	QSPI_M1_CS_N	I2S_S_WS	SPI_M_CS0_N	UART4_CTS	UART5_RX	PWM1_A	SPI_S_MOSI	GPIO_A10
GPIO_11	QSPI_M1_IO_3	I2S_S_TX_SDO	I2C3_SCL	UART4_RTS	UART5_TX	PWM1_B	SPI_S_CLK	GPIO_A11
GPIO_12	QSPI_M1_IO_2	I2S_S_RX_SDI	I2C3_SDA	UART4_TX	UART5_CTS	PWM1_C	SPI_S_MISO	GPIO_A12
GPIO_13	QSPI_M1_IO_1	I2S_S_SCLK	SPI_M_MISO	UART4_RX	UART5_RTS	FERP_GPIO_Trig	SPI_S_CS_N	GPIO_A13
GPIO_14	QSPI_M1_IO_0	I2C5_SCL	SPI_M_MOSI	PDM_CLKO	df_ant_switch_sel_2	UART0_TX	iso_sync1_p	GPIO_A14

IO Name	MUX_0	MUX_1	MUX_2	MUX_3	MUX_4	MUX_5	MUX_6	MUX_7
GPIO_15	QSPI_M1_CLK	I2C5_SDA	SPI_M_CLK	PDM_DI	df_ant_switch_sel_3	UART0_RX	iso_sync0_p	GPIO_A15

Table 8-3 Pin multiplexing for GPIO_16–GPIO_23

IO Name	MUX_0	MUX_1	MUX_2	MUX_3	MUX_4	MUX_5	MUX_6	MUX_7
GPIO_16	QSPI_M2_CLK	DC_CLK	DSPI_SCK	SPI_M_CLK	UART4_TX	PWM0_C	I2C0_SCL	GPIO_B0
GPIO_17	QSPI_M2_IO_0	DC_IO_0	DSPI_MOSI	SPI_M_MOSI	UART2_TX	UART3_CTS	I2C0_SDA	GPIO_B1
GPIO_18	QSPI_M2_IO_1	DC_IO_1	DSPI_MISO	SPI_M_MISO	UART2_RX	UART3_RTS	PWM1_A	GPIO_B2
GPIO_19	QSPI_M2_IO_2	DC_IO_2	DSPI_DCX	SPI_M_CS0_N	UART2_CTS	UART3_TX	PWM1_B	GPIO_B3
GPIO_20	QSPI_M2_IO_3	DC_IO_3	DSPI_CSS	SPI_M_CS1_N	UART2_RTS	UART3_RX	PWM1_C	GPIO_B4
GPIO_21	QSPI_M0_CLK	SPI_M_CLK	SIM_CLK	I2S_SCLK	I2S_S_SCLK	SPI_S_CLK	I2C3_SCL	GPIO_B5
GPIO_22	QSPI_M0_IO_0	SPI_M_MOSI	SIM_IO	I2S_RX_SDI	I2S_S_TX_SDO	SPI_S_MISO	I2C3_SDA	GPIO_B6
GPIO_23	QSPI_M0_IO_1	SPI_M_MISO	SIM_RST_N	I2S_TX_SDO	I2S_S_RX_SDI	SPI_S_MOSI	I2C2_SCL	GPIO_B7

Table 8-4 Pin multiplexing for GPIO_24–GPIO_33

IO Name	MUX_0	MUX_1	MUX_2	MUX_3	MUX_4	MUX_5	MUX_6	MUX_7
GPIO_24	QSPI_M0_IO_2	SPI_M_CS0_N	SIM_PRESENCE	I2S_WS	I2S_S_WS	SPI_S_CS_N	I2C2_SDA	GPIO_B8
GPIO_25	QSPI_M0_IO_3	SPI_M_CS1_N	FERP_GPIO_Trig	UART4_RX	I2C3_SCL	I2C1_SCL	PWM0_A	GPIO_B9
GPIO_26	QSPI_M0_CS_N	SPI_M_CS0_N	UART0_CTS	SPI_M_CLK	I2C3_SDA	I2C1_SDA	PWM0_B	GPIO_B10
GPIO_27	QSPI_M2_CS_N	DC_CS_N	UART0_RTS	SPI_M_MOSI	I2C2_SCL	I2C0_SCL	PWM0_C	GPIO_B11
GPIO_28	PDM_CLKO	iso_sync1_p	UART0_TX	SPI_M_MISO	I2C2_SDA	I2C0_SDA	SPI_M_CLK	GPIO_B12
GPIO_29	PDM_DI	iso_sync0_p	UART0_RX	SPI_M_CS0_N	FERP_GPIO_Trig	DC_DCX	SPI_M_MOSI	GPIO_B13
GPIO_30	SPI_S_MOSI	I2S_SCLK	I2S_S_WS	I2C1_SCL	UART1_CTS	I2C5_SCL	SPI_M_MISO	GPIO_B14
GPIO_31	SPI_S_CLK	I2S_RX_SDI	I2S_S_TX_SDO	I2C1_SDA	UART1_RTS	I2C5_SDA	SPI_M_CS0_N	GPIO_B15
GPIO_32	SPI_S_MISO	I2S_TX_SDO	I2S_S_RX_SDI	PDM_CLKO	UART1_TX	PWM0_B	I2C0_SCL	GPIO_C0
GPIO_33	SPI_S_CS_N	I2S_WS	I2S_S_SCLK	PDM_DI	UART1_RX	PWM0_A	I2C0_SDA	GPIO_C1

Table 8-5 Pin multiplexing for AON_GPIO_0–AON_GPIO_7

IO Name	MUX_0	MUX_1	MUX_2	MUX_3	MUX_4	MUX_5	MUX_6	MUX_7
AON_GPIO_0	UART4_CTS	I2C4_SCL	PWM0_A	SIM_PRESENCE	PWM0_C	UART2_TX		AON_GPIO_0
AON_GPIO_1	UART4_RTS	I2C4_SDA	PWM0_B	SIM_RST_N	PWM1_A	UART2_RX		AON_GPIO_1
AON_GPIO_2	UART4_TX	I2C1_SCL	PWM0_C	SIM_IO	PWM1_B	iso_sync_p		AON_GPIO_2
AON_GPIO_3	UART4_RX	I2C1_SDA	PWM1_A	SIM_CLK	PWM1_C			AON_GPIO_3
AON_GPIO_4	PWM0_A	UART5_RX	I2C0_SCL	PWM1_B	SIM_CLK	coex_ble_tx		AON_GPIO_4
AON_GPIO_5	PWM0_B	UART5_TX	I2C0_SDA	PWM1_C	iso_sync_p	coex_ble_rx		AON_GPIO_5

IO Name	MUX_0	MUX_1	MUX_2	MUX_3	MUX_4	MUX_5	MUX_6	MUX_7
AON_GPIO_6	PWM0_C	UART5_CTS	I2C5_SCL	PDM_CLKO	UART1_RX	coex_wlan_tx		AON_GPIO_6
AON_GPIO_7	PWM1_A	UART5_RTS	I2C5_SDA	PDM_DI	UART1_TX	coex_wlan_rx		AON_GPIO_7

Table 8-6 Pin multiplexing for MSIO_0–MSIO_7

IO Name	MUX_0	MUX_1	MUX_2	MUX_3	MUX_4	MUX_5	MUX_6	MUX_7
MSIO_0	PWM0_A	SIM_CLK	UART3_RX	I2S_SCLK	I2S_S_SCLK	PDM_DI		MSIO_A0
MSIO_1	PWM0_B	SIM_IO	UART3_TX	I2S_RX_SDI	I2S_S_RX_SDI	PDM_CLKO		MSIO_A1
MSIO_2	PWM0_C	SIM_RST_N	UART3_RTS	I2S_TX_SDO	I2S_S_TX_SDO	coex_wlan_rx		MSIO_A2
MSIO_3	PWM1_A	SIM_PRESENCE	UART3_CTS	I2S_WS	I2S_S_WS	coex_wlan_tx		MSIO_A3
MSIO_4	UART2_RTS	PWM1_B	I2C3_SDA		I2C0_SDA	coex_ble_rx	iso_sync0_p	MSIO_A4
MSIO_5	UART2_CTS	PWM1_C	I2C3_SCL		I2C0_SCL	coex_ble_tx	iso_sync1_p	MSIO_A5
MSIO_6	UART2_RX	PWM1_B	I2C4_SDA		I2C1_SDA	iso_sync0_p		MSIO_A6
MSIO_7	UART2_TX	PWM1_C	I2C4_SCL		I2C1_SCL	iso_sync1_p		MSIO_A7

8.1.3 Registers

8.1.3.1 DPAD_MUX_CTRL_00_07

- **Name:** GPIO 00_07 Mux Control Register
- **Description:** This register controls GPIO_0–GPIO_7 mux.
- **Base Address:** 0x4000EE00
- **Offset:** 0x040
- **Reset Value:** 0x77777700

Table 8-7 DPAD_MUX_CTRL_00_07 Register

Bits	Field Name	RW	Reset	Description
31:28	DPAD_MUX_SEL_07	RW	0x7	Value for GPIO_7 mux
27:24	DPAD_MUX_SEL_06	RW	0x7	Value for GPIO_6 mux
23:20	DPAD_MUX_SEL_05	RW	0x7	Value for GPIO_5 mux
19:16	DPAD_MUX_SEL_04	RW	0x7	Value for GPIO_4 mux
15:12	DPAD_MUX_SEL_03	RW	0x7	Value for GPIO_3 mux
11:8	DPAD_MUX_SEL_02	RW	0x7	Value for GPIO_2 mux
7:4	DPAD_MUX_SEL_01	RW	0x0	Value for GPIO_1 mux
3:0	DPAD_MUX_SEL_00	RW	0x0	Value for GPIO_0 mux

8.1.3.2 DPAD_MUX_CTRL_08_15

- **Name:** GPIO 08_15 Mux Control Register
- **Description:** This register controls GPIO_8–GPIO_15 mux.
- **Base Address:** 0x4000EE00
- **Offset:** 0x044
- **Reset Value:** 0x77777777

Table 8-8 DPAD_MUX_CTRL_08_15 Register

Bits	Field Name	RW	Reset	Description
31:28	DPAD_MUX_SEL_15	RW	0x7	Value for GPIO_15 mux
27:24	DPAD_MUX_SEL_14	RW	0x7	Value for GPIO_14 mux
23:20	DPAD_MUX_SEL_13	RW	0x7	Value for GPIO_13 mux
19:16	DPAD_MUX_SEL_12	RW	0x7	Value for GPIO_12 mux
15:12	DPAD_MUX_SEL_11	RW	0x7	Value for GPIO_11 mux
11:8	DPAD_MUX_SEL_10	RW	0x7	Value for GPIO_10 mux
7:4	DPAD_MUX_SEL_09	RW	0x7	Value for GPIO_9 mux
3:0	DPAD_MUX_SEL_08	RW	0x7	Value for GPIO_8 mux

8.1.3.3 DPAD_MUX_CTRL_16_23

- **Name:** GPIO 16_23 Mux Control Register
- **Description:** This register controls GPIO_16–GPIO_23 mux.
- **Base Address:** 0x4000EE00
- **Offset:** 0x048
- **Reset Value:** 0x77777777

Table 8-9 DPAD_MUX_CTRL_16_23 Register

Bits	Field Name	RW	Reset	Description
31:28	DPAD_MUX_SEL_23	RW	0x7	Value for GPIO_23 mux
27:24	DPAD_MUX_SEL_22	RW	0x7	Value for GPIO_22 mux
23:20	DPAD_MUX_SEL_21	RW	0x7	Value for GPIO_21 mux
19:16	DPAD_MUX_SEL_20	RW	0x7	Value for GPIO_20 mux
15:12	DPAD_MUX_SEL_19	RW	0x7	Value for GPIO_19 mux
11:8	DPAD_MUX_SEL_18	RW	0x7	Value for GPIO_18 mux
7:4	DPAD_MUX_SEL_17	RW	0x7	Value for GPIO_17 mux

Bits	Field Name	RW	Reset	Description
3:0	DPAD_MUX_SEL_16	RW	0x7	Value for GPIO_16 mux

8.1.3.4 DPAD_MUX_CTRL_24_31

- **Name:** GPIO 24_31 mux control register
- **Description:** This register controls GPIO_24–GPIO_31 mux.
- **Base Address:** 0x4000EE00
- **Offset:** 0x04C
- **Reset Value:** 0x77777777

Table 8-10 DPAD_MUX_CTRL_24_31 Register

Bits	Field Name	RW	Reset	Description
31:28	DPAD_MUX_SEL_31	RW	0x7	Value for GPIO_31 mux
27:24	DPAD_MUX_SEL_30	RW	0x7	Value for GPIO_30 mux
23:20	DPAD_MUX_SEL_29	RW	0x7	Value for GPIO_29 mux
19:16	DPAD_MUX_SEL_28	RW	0x7	Value for GPIO_28 mux
15:12	DPAD_MUX_SEL_27	RW	0x7	Value for GPIO_27 mux
11:8	DPAD_MUX_SEL_26	RW	0x7	Value for GPIO_26 mux
7:4	DPAD_MUX_SEL_25	RW	0x7	Value for GPIO_25 mux
3:0	DPAD_MUX_SEL_24	RW	0x7	Value for GPIO_24 mux

8.1.3.5 AON_PAD_MUX_CTRL

- **Name:** AON GPIO Pin Mux Control Register
- **Description:** This register controls the AON GPIO pin mux.
- **Base Address:** 0x4000EE00
- **Offset:** 0x050
- **Reset Value:** 0x77777777

Table 8-11 AON_PAD_MUX_CTRL Register

Bits	Field Name	RW	Reset	Description
31	RSVD	R		Reserved bits
30:28	AON_PAD_MUX_SEL_07	RW	0x7	Value for AON_GPIO_7 mux
27	RSVD	R		Reserved bits
26:24	AON_PAD_MUX_SEL_06	RW	0x7	Value for AON_GPIO_6 mux

Bits	Field Name	RW	Reset	Description
23	RSVD	R		Reserved bits
22:20	AON_PAD_MUX_SEL_05	RW	0x7	Value for AON_GPIO_5 mux
19	RSVD	R		Reserved bits
18:16	AON_PAD_MUX_SEL_04	RW	0x7	Value for AON_GPIO_4 mux
15	RSVD	R		Reserved bits
14:12	AON_PAD_MUX_SEL_03	RW	0x7	Value for AON_GPIO_3 mux
11	RSVD	R		Reserved bits
10:8	AON_PAD_MUX_SEL_02	RW	0x7	Value for AON_GPIO_2 mux
7	RSVD	R		Reserved bits
6:4	AON_PAD_MUX_SEL_01	RW	0x7	Value for AON_GPIO_1 mux
3:0	RSVD	R		Reserved bits

8.1.3.6 MSIO_A_PAD_MUX_CTRL

- **Name:** MSIOA Pin Mux Control Register
- **Description:** This register controls the MSIO pin mux.
- **Base Address:** 0x4000EE00
- **Offset:** 0x054
- **Reset Value:** 0x77777777

Table 8-12 MSIO_A_PAD_MUX_CTRL Register

Bits	Field Name	RW	Reset	Description
31	RSVD	R		Reserved bits
30:28	MSIO_A_MUX_SEL_07	RW	0x7	Value for MSIO_7 mux
27	RSVD	R		Reserved bits
26:24	MSIO_A _MUX_SEL_06	RW	0x7	Value for MSIO_6 mux
23	RSVD	R		Reserved bits
22:20	MSIO_A _MUX_SEL_05	RW	0x7	Value for MSIO_5 mux
19	RSVD	R		Reserved bits
18:16	MSIO_A _MUX_SEL_04	RW	0x7	Value for MSIO_4 mux
15	RSVD	R		Reserved bits
14:12	MSIO_A _MUX_SEL_03	RW	0x7	Value for MSIO_3 mux
11	RSVD	R		Reserved bits
10:8	MSIO_A _MUX_SEL_02	RW	0x7	Value for MSIO_2 mux
7	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
6:4	MSIO_A _MUX_SEL_01	RW	0x7	Value for MSIO_1 mux
3	RSVD	R		Reserved bits
2:0	MSIO_A _MUX_SEL_00	RW	0x7	Value for MSIO_0 mux

8.1.3.7 DPAD_MUX_CTRL_32_33

- **Name:** GPIO 32_33 Mux Control Register
- **Description:** This register controls the GPIO_32– GPIO_33 mux.
- **Base Address:** 0x4000EE00
- **Offset:** 0x058
- **Reset Value:** 0x00000077

Table 8-13 DPAD_MUX_CTRL_32_33 Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:4	DPAD_MUX_SEL_33	RW	0x7	Value for GPIO_33 mux
3:0	DPAD_MUX_SEL_32	RW	0x7	Value for GPIO_32 mux

8.2 I/O

The GPIOs of a GR5526 SoC can be connected to various digital and mixed-signal interfaces, and the features are as follows:

- Up to 34 General Purpose I/Os (GPIOs): normal digital GPIOs in MCU domain
- 8 AON_GPIOs: digital GPIOs in the AON domain
- 8 MSIOs: digital and analog mixed-signal GPIOs, no interrupt
- All I/Os have internal pull up/down registers
- All I/Os except fast speed I/Os (QSPI, OSPI, and XQSPI) can retain their last state when the system enters the sleep mode.
- GPIO_0–GPIO_15 are supplied by VDDIO_1.
- MSIO_A0–MSIO_A7 are supplied by VBATL.
- AON_GPIO_0–AON_GPIO_7 and GPIO_16–GPIO_33 are supplied by VDDIO_0.

8.2.1 GPIO

8.2.1.1 Introduction

A GR5526 SoC has up to 34 individually configurable GPIOs that can serve as peripherals or digital input/output.

8.2.1.2 Main Features

- Up to 34 digital GPIOs: GPIO_0–GPIO_15 are supplied by VDDIO_1. GPIO_16–GPIO_33 are supplied by VDDIO_0.
- All I/Os except GPIO_10–GPIO_25, GPIO_32, and GPIO_33 can retain their output levels when the system enters sleep mode.
- GPIO default state is input mode with pull-down.
- All GPIOs can be configured as being triggered by high level, low level, rising edge, falling edge, or both edges.

8.2.1.3 Functional Description

The 34 digital GPIOs can be used:

- For general-purpose input/output.
- To multiplex different peripherals on different output pins of the package.
- To multiplex debug signals from different blocks of the GR5526 SoC.

A functional drawing of the GPIO pads is provided in Figure 8-1. The digital GPIO pads can be configured to set direction, enable pull-up/pull-down resistors, and enable output retention. GPIOs can also be read or written by firmware for applications that need direct access through the GPIOx registers.

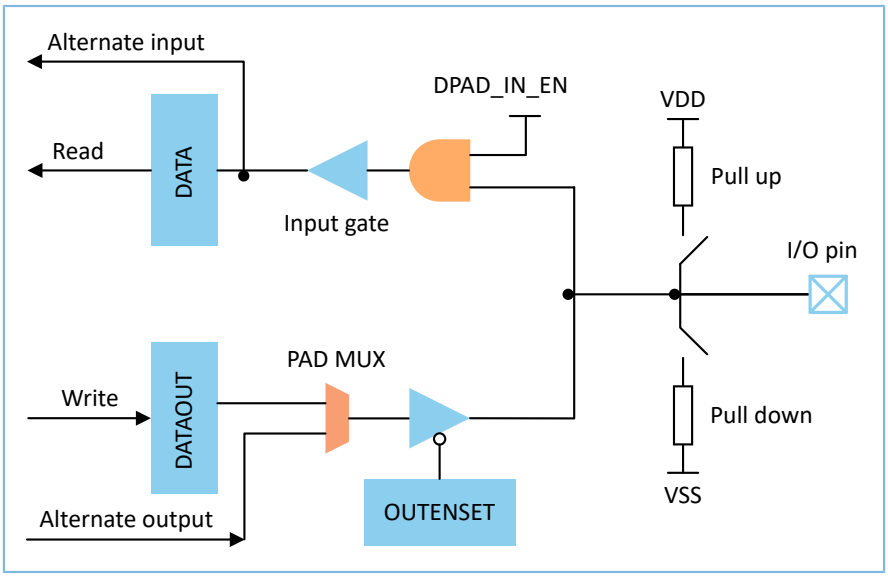

Figure 8-1 GPIO pad diagram

The general rule for enabling the on-chip pull-up/pull-down resistors is as follows:

- If a GPIO is unused, enable a pull-down resistor.
- If a GPIO is configured as an output, disable the pull-up/pull-down resistor. However, if disabling the output causes a high impedance state, enable the pull-up or pull-down resistor during this period.

- If a GPIO is used as an input and is being actively driven high or low by an external device, disable the pull-up/pull-down resistor (the same thing goes when there is an external pull-up/pull-down resistor on the board. In this case, the on-chip resistor should always be disabled). However, if the external device driving the input ever goes to a high impedance state, a pull-up or pull-down resistor must be enabled during this period.

Note:

GPIO_10–GPIO_25 become high impedance in sleep and cannot retain their output levels. During sleep, the signal level of these IOs is determined by the pull-up/pull-down resistors.

8.2.1.4 Registers

8.2.1.4.1 DPAD_R_EN_BUS_0

- **Name:** GPIO Pull-Up/Down Resistor Enable Register
- **Description:** This register decides whether to use pull-up/down resistors.
- **Base Address:** 0x4000EE00
- **Offset:** 0x000
- **Reset Value:** 0x00000000

Table 8-14 DPAD_R_EN_BUS_0 Register

Bits	Field Name	RW	Reset	Description
31:0	DPAD_R_EN	RW	0x0	Resistor enable control of GPIO_31–GPIO_0 **Value**: • 0x0: Enable pull-up/down resistor. • 0x1: Disable pull-up/down resistor.

8.2.1.4.2 DPAD_RTYP_BUS_0

- **Name:** GPIO Resistor Type Control Register
- **Description:** This register sets the resistor type to either pull-up or pull-down.
- **Base Address:** 0x4000EE00
- **Offset:** 0x004
- **Reset Value:** 0x00000000

Table 8-15 DPAD_RTYP_BUS_0 Register

Bits	Field Name	RW	Reset	Description
31:0	DPAD_RTYP	RW	0x0	Resistor type of GPIO_31–GPIO_0 **Value**:

Bits	Field Name	RW	Reset	Description
				• 0x0: Pull down. • 0x1: Pull up.

8.2.1.4.3 DPAD_IN_EN_BUS_0

- **Name:** GPIO Input Enable Register
- **Description:** This register enables a GPIO as input.
- **Base Address:** 0xA000E000
- **Offset:** 0x008
- **Reset Value:** 0x00000000

Table 8-16 DPAD_IN_EN_BUS_0 Register

Bits	Field Name	RW	Reset	Description
31:0	DPAD_IN_EN	RW	0x0	Active I/O inputs to GPIO_31–GPIO_0 **Value**: • 0x0: Enable inputs to pad. • 0x1: Disable inputs to pad.

8.2.1.4.4 DPAD_R_EN_BUS_1

- **Name:** GPIO Pull-Up/Down Resistor Enable Register
- **Description:** This register decides whether to use pull-up/down resistors.
- **Base Address:** 0x4000EE00
- **Offset:** 0x000
- **Reset Value:** 0x00000000

Table 8-17 DPAD_R_EN_BUS_1 Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1:0	DPAD_R_EN	RW	0x0	Resistor enable control of GPIO_33–GPIO_32. **Value**: • 0x0: Enable pull-up/down resistor. • 0x1: Disable pull-up/down resistor.

8.2.1.4.5 DPAD_RTYP_BUS_1

- **Name:** GPIO Pull-Up/Down Resistor Type Control Register

- **Description:** This register sets the resistor type to either pull-up or pull-down.
- **Base Address:** 0x4000EE00
- **Offset:** 0x004
- **Reset Value:** 0x00000000

Table 8-18 DPAD_RTYP_BUS_1 Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1:0	DPAD_RTYP	RW	0x0	Resistor type of GPIO_33–GPIO_32 **Value**: • 0x0: Pull down. • 0x1: Pull up.

8.2.1.4.6 DPAD_IN_EN_BUS_1

- **Name:** GPIO Input Enable Register
- **Description:** This register enables a GPIO as input.
- **Base Address:** 0x4000E000
- **Offset:** 0x008
- **Reset Value:** 0x00000000

Table 8-19 DPAD_IN_EN_BUS_1 Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1:0	DPAD_IN_EN	RW	0x0	Active I/O inputs to the GPIO_33 – GPIO_32 **Value**: • 0x0: Enable inputs to pad. • 0x1: Disable inputs to pad.

8.2.1.4.7 DPAD_R_EN_MISC

- **Name:** OSPI and XQSPI I/O Pull-Up/Down Resistor Enable Register
- **Description:** This register decides whether to use pull-up/down resistors.
- **Base Address:** 0x4000EE00
- **Offset:** 0x000
- **Reset Value:** 0x00000000

Table 8-20 DPAD_R_EN_MISC Register

Bits	Field Name	RW	Reset	Description
31:22	RSVD	R		Reserved bits
21:18	xqspi_io	RW	0x0	**Value**: • 0x0: Enable pull-up/down resistor for XQSPI I/O. • 0x1: Disable pull-up/down resistor for XQSPI I/O.
17	xqspi_cs	RW	0x0	**Value**: • 0x0: Enable pull-up/down resistor for XQSPI CS. • 0x1: Disable pull-up/down resistor for XQSPI CS.
16	xqspi_clk	RW	0x0	**Value**: • 0x0: Enable pull-up/down resistor for XQSPI clk. • 0x1: Disable pull-up/down resistor for XQSPI clk.
15:11	RSVD	R		Reserved bits
10:3	ospi_io	RW	0x0	**Value**: • 0x0: Enable pull-up/down resistor for OSPI I/O. • 0x1: Disable pull-up/down resistor for OSPI I/O.
2	ospi_cs	RW	0x0	**Value**: • 0x0: Enable pull-up/down resistor for OSPI CS. • 0x1: Disable pull-up/down resistor for OSPI CS.
1	ospi_dqs	RW	0x0	**Value**: • 0x0: Enable pull-up/down resistor for OSPI DQS. • 0x1: Disable pull-up/down resistor for OSPI DQS.
0	ospi_clk	RW	0x0	**Value**: • 0x0: Enable pull-up/down resistor for OSPI clk. • 0x1: Disable pull-up/down resistor for OSPI clk.

8.2.1.4.8 DPAD_RTYP_MISC

- **Name:** OSPI and XQSPI Pull-Up/Down Resistor Type Control Register
- **Description:** This register sets the resistor type to either pull-up or pull-down.
- **Base Address:** 0x4000EE00
- **Offset:** 0x004
- **Reset Value:** 0x003F0000

Table 8-21 DPAD_RTYP_MISC Register

Bits	Field Name	RW	Reset	Description
31:22	RSVD	R		Reserved bits
21:18	xqspi_io	RW	0x0	**Value**: • 0x0: Pull down. • 0x1: Pull up.
17	xqspi_cs	RW	0x0	**Value**: • 0x0: Pull down. • 0x1: Pull up.
16	xqspi_clk	RW	0x0	**Value**: • 0x0: Pull down. • 0x1: Pull up.
15:11	RSVD	R		Reserved bits
10:3	ospi_io	RW	0x0	**Value**: • 0x0: Pull down. • 0x1: Pull up.
2	Ospi_cs	RW	0x0	**Value**: • 0x0: Pull down. • 0x1: Pull up.
1	ospi_dqs	RW	0x0	**Value**: • 0x0: Pull down. • 0x1: Pull up.
0	ospi_clk	RW	0x0	**Value**: • 0x0: Pull down. • 0x1: Pull up.

8.2.1.4.9 DPAD_IN_EN_MISC

- **Name:** OSPI and XQSPI Pin Input Enable Register
- **Description:** This register enables the input.
- **Base Address:** 0x4000E000
- **Offset:** 0x008
- **Reset Value:** 0x00000000

Table 8-22 DPAD_IN_EN_MISC Register

Bits	Field Name	RW	Reset	Description
31:22	RSVD	R		Reserved bits
21:18	xqspi_io	RW	0x0	**Value**: • 0x0: Enable inputs to pad for XQSPI I/O. • 0x1: Disable inputs to pad for XQSPI I/O.
17	xqspi_cs	RW	0x0	**Value**: • 0x0: Enable inputs to pad for XQSPI CS. • 0x1: Disable inputs to pad for XQSPI CS.
16	xqspi_clk	RW	0x0	**Value**: • 0x0: Enable inputs to pad for XQSPI clk. • 0x1: Disable inputs to pad for XQSPI clk.
15:11	RSVD	R		Reserved bits
10:3	ospi_io	RW	0x0	**Value**: • 0x0: Enable inputs to pad for OSPI I/O. • 0x1: Disable inputs to pad for OSPI I/O.
2	ospi_cs	RW	0x0	**Value**: • 0x0: Enable inputs to pad for OSPI CS. • 0x1: Disable inputs to pad for OSPI CS.
1	ospi_dqs	RW	0x0	**Value**: • 0x0: Enable inputs to pad for OSPI DQS. • 0x1: Disable inputs to pad for OSPI DQS.
0	ospi_clk	RW	0x0	**Value**: • 0x0: Enable inputs to pad for OSPI clk. • 0x1: Disable inputs to pad for OSPI clk.

8.2.1.4.10 GPIOx_DATA

- **Name:** GPIO Data Value Register
- **Description:** This register contains the input data and can write it to data output register.
- **Base Address:** 0x40010000 + x*0x1000
- **Offset:** 0x0000
- **Reset Value:** 0x00000000

Table 8-23 GPIOx_DATA Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	DATA	RW	0x0	Data value [15:0]: • Read: Read the value that is sampled at pin • Write: Write the field to the GPIOx_DATA_OUT register.

Note:

GPIOx includes GPIOA (GPIO_0–GPIO_15), GPIOB (GPIO_16–GPIO_31), and GPIOC (GPIO_32–GPIO_33).

8.2.1.4.11 GPIOx_DATA_OUT

- **Name:** GPIO Data Output Register
- **Description:** This register contains the output data.
- **Base Address:** 0x40010000 + x*0x1000
- **Offset:** 0x0004
- **Reset Value:** 0x00000000

Table 8-24 GPIOx_DATA_OUT Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	DATA_OUT	RW	0x0	Data output register value [15:0]: • Read: current value of data output register • Write: to data output register

8.2.1.4.12 GPIOx_OUT_EN

- **Name:** GPIO Output Enable Register
- **Description:** This register is used to enable the GPIO as output.
- **Base Address:** 0x40010000 + x*0x1000
- **Offset:** 0x0010
- **Reset Value:** 0x00000000

Table 8-25 GPIOx_OUT_EN Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	EN	RW	0x0	Output enable set [15:0]:

Bits	Field Name	RW	Reset	Description
				Read: • 0x0: As input • 0x1: As output **Write**: • 0x0: No effect • 0x1: Set the output enable bit.

8.2.1.4.13 GPIOx_OUT_CLR

- **Name:** GPIO Output Enable Clear Register
- **Description:** This register is used to disable output bit.
- **Base Address:** 0x40010000 + x*0x1000
- **Offset:** 0x0014
- **Reset Value:** 0x00000000

Table 8-26 GPIOx_OUT_CLR Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	CLR	RW	0x0	Output enable clear [15:0]: **Read**: • 0x0: as input • 0x1: as output **Write**: • 0x0: No effect • 0x1: Clear the output enable bit.

8.2.1.4.14 GPIOx_INT_EN

- Name: GPIO Interrupt Enable Register
- Description: This register is used to enable GPIO interrupt.
- Base Address: 0x40010000 + x*0x1000
- Offset: 0x0020
- Reset Value: 0x00000000

Table 8-27 GPIOx_INT_EN Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	EN	RW	0x0	Interrupt enable set [15:0]: **Read**: • 0x0: Interrupt disabled • 0x1: Interrupt enabled **Write**: • 0x0: No effect • 0x1: Set the interrupt enable bit.

8.2.1.4.15 GPIOx_INT_CLR

- **Name:** GPIO Interrupt Enable Clear Register
- **Description:** This register is used to disable GPIO interrupt.
- **Base Address:** 0x40010000 + x*0x1000
- **Offset:** 0x0024
- **Reset Value:** 0x00000000

Table 8-28 GPIOx_INT_CLR Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	CLR	RW	0x0	Interrupt enable clear [15:0]: **Read**: • 0x0: Interrupt disabled • 0x1: Interrupt enabled **Write**: • 0x0: No effect • 0x1: Clear the interrupt enable bit.

8.2.1.4.16 GPIOx_INT_TYPE_EN

- **Name:** GPIO Interrupt Type Enable Register
- **Description:** This register is used to set GPIO interrupt type.
  - Interrupt type is 0 and interrupt polarity is 0: low-level trigger interrupt

- Interrupt type is 0 and interrupt polarity is 1: high-level trigger interrupt
- Interrupt type is 1 and interrupt polarity is 0: falling edge trigger interrupt
- Interrupt type is 1 and interrupt polarity is 1: rising edge trigger interrupt

- **Base Address:** 0x40010000 + x*0x1000
- **Offset:** 0x0028
- **Reset Value:** 0x00000000

Table 8-29 GPIOx_INT_TYPE_EN Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	EN	RW	0x0	Interrupt type set [15:0]: **Read**: • 0x0: Low or high level • 0x1: Falling edge or rising edge **Write**: • 0x0: No effect • 0x1: Set the interrupt type bit.

8.2.1.4.17 GPIOx_INT_TYPE_CLR

- **Name:** GPIO Interrupt Type Clear Register
- **Description:** This register is used to disable GPIO interrupt type.
- **Base Address:** 0x40010000 + x*0x1000
- **Offset:** 0x002C
- **Reset Value:** 0x00000000

Table 8-30 GPIOx_INT_TYPE_CLR Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	CLR	RW	0h 0	Interrupt type clear [15:0]: **Read**: • 0x0: Low or high level • 0x1: Falling edge or rising edge **Write**: • 0x0: No effect

Bits	Field Name	RW	Reset	Description
				• 0x1: Clear the interrupt type bit.

8.2.1.4.18 GPIOx_INT_POL_EN

- **Name:** GPIO Interrupt Polarity Enable Register
- **Description:** This register is used to enable GPIO interrupt polarity.
- **Base Address:** 0x40010000 + x*0x1000
- **Offset:** 0x0030
- **Reset Value:** 0x00000000

Table 8-31 GPIOx_INT_POL_EN Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	EN	RW	0h 0	Polarity configuration [15:0]: **Read**: • 0x0: Low level or falling edge • 0x1: High level or rising edge **Write**: • 0x0: No effect • 0x1: Set the interrupt polarity bit.

8.2.1.4.19 GPIOx_INT_POL_CLR

- **Name:** GPIO Interrupt Polarity Disable Register
- **Description:** This register is used to disable GPIO interrupt polarity.
- **Base Address:** 0x40010000 + x*0x1000
- **Offset:** 0x0034
- **Reset Value:** 0x00000000

Table 8-32 GPIOx_INT_POL_CLR Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	CLR	RW	0x0	Polarity configuration [15:0]: **Read**: • 0x0: Low level or falling edge

Bits	Field Name	RW	Reset	Description
				• 0x1: High level or rising edge **Write**: • 0x0: No effect • 0x1: Clear the interrupt polarity bit.

8.2.1.4.20 GPIOx_INT_STAT

- **Name:** GPIO IRQ Status Register
- **Description:** This register contains GPIO interrupt status and is used to clear the interrupt request.
- **Base Address:** 0x40010000 + x*0x1000
- **Offset:** 0x0038
- **Reset Value:** 0x00000000

Table 8-33 GPIOx_INT_STAT Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	STAT_CLR	RW	0x0	Write one to clear interrupt request: **Read**: • [15:0] IRQ status register **Write**: • 0x0: No effect • 0x1: To clear the interrupt request

8.2.1.5 Electrical Specifications

The electrical parameters for the GPIO_0–GPIO_33 are as follows:

Table 8-34 GPIO electrical specifications

Parameter	Description	Min.	Typ.	Max.	Unit
V_{IH}	Input high voltage	VDDIO x 0.7		VDDIO	V
V_{IL}	Input low voltage	VSSIO		VDDIO x 0.3	V
$V_{OH,L}$	Output high voltage, 4 mA, VDDIO ≥ 1.7 V	VDDIO – 0.4		VDDIO	V
$V_{OH,M}$	Output high voltage, 4 mA, VDDIO ≥ 2.5 V	VDDIO – 0.4		VDDIO	V
VOH,H	Output high voltage, 4 mA, VDDIO ≥ 3 V	VDDIO – 0.4		VDDIO	V
$V_{OL,L}$	Output low voltage, 4 mA, VDDIO ≥1.7 V	VSS		VSS + 0.4	V

Parameter	Description	Min.	Typ.	Max.	Unit
$V_{OL,M}$	Output low voltage, 4 mA, VDDIO ≥ 2.5 V	VSS		VSS + 0.4	V
VOL,H	Output low voltage, 4 mA, VDDIO ≥ 3 V	VSS		VSS + 0.4	V
$I_{OL,L}$	Current at VSS+0.4 V, output set low, VDDIO ≥1.7 V			4	mA
$I_{OL,M}$	Current at VSS+0.4 V, output set low, VDDIO ≥ 2.5 V			5	mA
$I_{OH,L}$	Current at VDD-0.4 V, output set high, VDDIO ≥ 1.7 V			4	mA
$I_{OH,M}$	Current at VDD-0.4 V, output set high, VDDIO ≥ 2.5 V			5	mA
$t_{RF,15pF}$	Rise/Fall time, 10% – 90%, 15 pF load		5	8	ns
$t_{RF,25pF}$	Rise/Fall time, 10% – 90%, 25 pF load		6	13	ns
R_{PU}	Pull-up resistance	70	120	150	kΩ
R_{PD}	Pull-down resistance	70	120	150	kΩ
C_{PAD}	Pad capacitance		5		pF

8.2.2 Always-On I/O

8.2.2.1 Introduction

A GR5526 SoC has eight individually configurable AON_GPIOs that can be applied by peripherals or digital input/output.

8.2.2.2 Functional Description

For all AON_GPIOs,

- The I/O default state is input mode with pull-down.
- Can be configured as being triggered by high level, low level, rising edge, falling edge, or both edges
- Can work as wake-up sources from deep sleep.
- Can generate slow-speed clocks.
- Can work as digital GPIOs for input/output.
- Can be multiplexed for different peripherals.

8.2.2.3 Registers

8.2.2.3.1 AON_PAD_CTRL0

- **Name:** Always-on Pad Control Register
- **Description:** This register contains the AON_GPIO configurations.
- **Base Address:** 0x4000A000

- **Offset:** 0x1E8
- **Reset Value:** 0x00000000

Table 8-35 AON_PAD_CTRL_0 Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:8	AON_R_TYPE	RW	0x0	Always-on PAD resistor type **Value**: • 0x0: Pull down. • 0x1: Pull up.
7:0	AON_RE_N	RW	0x0	Always-on PAD resister enable **Value**: • 0x0: Enable AONx resistor. • 0x1: Disable AONx resistor.

8.2.2.3.2 AON_PAD_CTRL1

- **Name**: Always-on Pad Control Register
- **Description**: This register contains the input and output data.
- **Base Address**: 0x4000A000
- **Offset**: 0x1EC
- **Reset Value**: 0x000000FF

Table 8-36 AON_PAD_CTRL_1 Register

Bits	Field Name	RW	Reset	Description
31:24	RSVD	R		Reserved bits
23:16	AON_INPUT_VAL	RW	0x0	AON pad input value (valid when oe_n = 1) **Value**: • 0x0: input AONx low. • 0x1: input AONx high.
15:8	AON_OUT_VAL	RW	0x0	AON PAD output value (valid when oe_n = 0) **Value**: • 0x0: Drive AONx low. • 0x1: Drive AONx high.
7:0	AON_O_N	RW	0xFF	Always-on pad output enable (active low) **Value**:

Bits	Field Name	RW	Reset	Description
				• 0x0: Enable AONx output. • 0x1: Disable AONx output.

8.2.2.3.3 AON_PAD_CLK

- **Name:** Always-on PAD output clock controls register
- **Description:** This register controls inner clock output from AON_GPIO_4.
- **Base Address:** 0x4000A000
- **Offset:** 0x1F0
- **Reset Value:** 0x00000000

Table 8-37 AON_PAD_CLK Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4:2	CLK_OUT_SEL	RW	0x0	Clock out selection **Value:** • 0x00: RNG_OSC (RNG Osc 32 kHz Clock) • 0x01: HFXO_32K (HFXO 32 kHz derived from HFXO_32 MHz Clock) • 0x02: LFRC_32K (LFRC 32 kHz Clock)[Z1] • 0x03: LFXO_32K (LF_XO 32 kHz Clock)
1	RSVD	R		Reserved bits
0	CLK_OUT_EN	RW	0x0	Enable clock out via AON GPIO 4. **Value:** • 0x0: Disable selected clock out via AON GPIO 4. • 0x1: Enable selected clock out via AON GPIO 4.

8.2.2.3.4 AON_PAD_MCU_OVR

- **Name**: Always-on Pad Control Register
- **Description**: This register controls the MCU domain setting of always-on pads.
- **Base Address**: 0x4000A000
- **Offset**: 0x1F4
- **Reset Value**: 0x000000FF

Table 8-38 AON_PAD_MCU_OVR Register

Bits	Field Name	RW	Reset	Description
31:17	RSVD	R		Reserved bits
23:16	AON_PAD_MCU_OVR	RW	0x0	Use the settings from MCU domain; only valid when MCU domain is ON • 0x0: Disable AONx digital setting. • 0x1: Enable AONx digital setting.
15:0	RSVD	R		Reserved bits

8.2.2.4 Electrical Specifications

The electrical parameters for the AON_GPIO_0–AON_GPIO_7 are as follows:

Table 8-39 GPIO electrical specifications

Parameter	Description	Min.	Typ.	Max.	Unit
V_{IH}	Input high voltage	VDDIO x 0.7		VDDIO	V
V_{IL}	Input low voltage	VSSIO		VDDIO x 0.3	V
$V_{OH,L}$	Output high voltage, 4 mA, VDDIO ≥ 1.7 V	VDDIO – 0.4		VDDIO	V
$V_{OH,M}$	Output high voltage, 4 mA, VDDIO ≥ 2.5 V	VDDIO – 0.4		VDDIO	V
$V_{OH,H}$	Output high voltage, 4 mA, VDDIO ≥ 3 V	VDDIO – 0.4		VDDIO	V
$V_{OL,L}$	Output low voltage, 4 mA, VDDIO ≥ 1.7 V	VSS		VSS + 0.4	V
$V_{OL,M}$	Output low voltage, 4 mA, VDDIO ≥ 2.5 V	VSS		VSS + 0.4	V
$V_{OL,H}$	Output low voltage, 4 mA, VDDIO ≥ 3 V	VSS		VSS + 0.4	V
$I_{OL,L}$	Current at VSS+0.4 V, output set low, VDDIO ≥1.7 V			4	mA
$I_{OL,M}$	Current at VSS+0.4 V, output set low, VDDIO ≥ 2.5 V			5	mA
$I_{OH,L}$	Current at VDD-0.4 V, output set high, VDDIO ≥ 1.7 V			4	mA
$I_{OH,M}$	Current at VDD-0.4 V, output set high, VDD ≥ 2.5 V			5	mA
$t_{RF,15pF}$	Rise/Fall time, 10%–90%, 15 pF load		5	8	ns
$t_{RF,25pF}$	Rise/Fall time, 10%–90%, 25 pF load		6	13	ns
R_{PU}	Pull-up resistance	70	120	150	kΩ
R_{PD}	Pull-down resistance	70	120	150	kΩ
C_{PAD}	Pad capacitance		5		pF

8.2.3 Mixed Signal I/O

8.2.3.1 Introduction

A GR5526 SoC has eight individually configurable MSIOs that can be applied by peripherals, digital input\output or analog input.

8.2.3.2 Functional Description

- MSIO default state is analog input mode.
- As analog input for sense ADC or Comparator.
- As digital MSIOs for input/output, no interrupt.
- To multiplex some different peripherals.
- For digital modules with high timing requirements, it is recommended to increase the MSIO drive current.

8.2.3.3 Registers

8.2.3.3.1 MSIO_A_PAD_CFG0

- **Name:** MSIOA Pad Control Register
- **Description:** This register contains the input and output data.
- **Base Address:** 0x 4000A1D8
- **Offset:** 0x00
- **Reset Value:** 0x00000000

Table 8-40 MSIO_A_PAD_CFG0 Register

Bits	Field Name	RW	Reset	Description
31:24	RSVD	R		Reserved bits
23:16	MSIO_A_VAL	R	0x0	MSIOA digital value
15:8	MSIO_A_OE_N	RW	0xFF	MSIOA output enable (active low) **Value**: • 0x0: Enable MSIOA_x output. • 0x1: Disable MSIOA_x output.
7:0	MSIO_A_OUT	RW	0x0	MSIOA output value (valid in output mode)

8.2.3.3.2 MSIO_A_PAD_CFG1

- **Name:** MSIOA Pad Control Register
- **Description:** This register contains the MSIOA configurations.
- **Base Address:** 0x 4000A1DC
- **Offset:** 0x00
- **Reset Value:** 0x00FF0000

Table 8-41 MSIO_A_PAD_CFG1 Register

Bits	Field Name	RW	Reset	Description
31:24	MSIO_A_IE_N	RW	0x0	MSIOA input enable (active low) **Value**: • 0x0: Enable input. • 0x1: Disable input.
23:16	MSIO_A_RE_N	RW	0xFF	MSIOA resistor enable (active low) **Value**: • 0x0: Enable resistor. • 0x1: Disable resistor.
15:8	MSIO_A_RTYPE	RW	0x0	MSIOA resistor type **Value**: • 0x0: Pull down. • 0x1: Pull up.
7:0	MSIO_A_AE_N	RW	0x0	Analog enable control for MSIOA pad **Value**: • 0x0: analog mode • 0x1: digital mode

8.2.3.3.3 MSIO_A_PAD_MCU_OVR

- **Name:** MSIOA Using MCU Domain Setting Control Register
- **Description:** This register controls MSIOA using the settings from the MCU domain.
- **Base Address:** 0x4000A1E0
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 8-42 MSIO_A_PAD_MCU_OVR Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R	N/A	Reserved bits
7:0	msio_a_mcu_ovr	RW	0x0	Use the settings from MCU domain, only valid when MCU domain is ON **Value**: • 0x0: Use settings in MSIO_A_PAD_CFG as GPIO. • 0x1: Use PAD MUX settings in MCU.

8.2.3.4 Electrical Specifications

The electrical parameters for the MSIO_A0–MSIO_A7 are as follows:

Table 8-43 GPIO electrical specifications

Parameter	Description	Min.	Typ.	Max.	Unit
V_{IH}	Input high voltage	VDDIO x 0.7		VDDIO	V
V_{IL}	Input low voltage	VSSIO		VDDIO x 0.3	V
$V_{OH,L}$	Output high voltage, 2.5 mA, VDDIO ≥ 1.7 V	VDDIO – 0.4		VDDIO	V
$V_{OH,M}$	Output high voltage, 2.5 mA, VDDIO ≥ 2.5 V	VDDIO – 0.4		VDDIO	V
$V_{OH,H}$	Output high voltage, 2.5 mA, VDDIO ≥ 3 V	VDDIO – 0.4		VDDIO	V
$V_{OL,L}$	Output low voltage, 2.5 mA, VDDIO ≥ 1.7 V	VSS		VSS + 0.4	V
$V_{OL,M}$	Output low voltage, 2.5 mA, VDDIO ≥ 2.5 V	VSS		VSS + 0.4	V
$V_{OL,H}$	Output low voltage, 2.5 mA, VDDIO ≥ 3 V	VSS		VSS + 0.4	V
$I_{OL,L}$	Current at VSS+0.4 V, output set low, VDDIO ≥1.7 V			2.5	mA
$I_{OL,M}$	Current at VSS+0.4 V, output set low, VDDIO ≥ 2.5 V			2.5	mA
$I_{OH,L}$	Current at VDD-0.4 V, output set high, VDDIO ≥ 1.7 V			2.5	mA
$I_{OH,M}$	Current at VDD-0.4 V, output set high, VDDIO ≥ 2.5 V			2.5	mA
$t_{RF,15pF}$	Rise/Fall time, 10%–90%, 15 pF load		5	8	ns
$t_{RF,25pF}$	Rise/Fall time, 10%–90%, 25 pF load		6	13	ns
R_{PU}	Pull-up resistance	70	120	150	kΩ
RPD	Pull-down resistance	70	120	150	kΩ
C_{PAD}	Pad capacitance		5		pF

8.3 Timer Interfaces

8.3.1 Timer

8.3.1.1 Introduction

The GR5526 SoC has two instances of Timer module that is a 32-bit general-purpose timer. Since the Timer is in the MCU Subsystem domain, Timer interrupt is not available during sleep, and the running Timer will not prevent MCU from sleeping. In active mode, when Timer is started again after stopping, it will continue to count down from the stop time. The value of RELOAD register will not take effect until the end of the current cycle.

8.3.1.2 Main Features

- Generate an interrupt when the 32-bit down counter reaches 0.
- The clock source of the Timer follows the system clock.

- Each timer module has an independent interrupt entry.
- Timer pauses when MCU is suspended.

8.3.1.3 Functional Description

- The Timer can generate an interrupt request signal when the 32-bit down counter reaches 0. The interrupt request is held until it is cleared by writing **1** to the INT_STAT register.
- If the Timer count reaches 0 and, at the same time, the software clears a previous interrupt status, the interrupt status is set to **1**.
- When the value of the RELOAD register is configured to **0**, no count and no interrupt are generated.
- The RELOAD value only represents the count-down number of timers. If you want to configure a specific duration (for example, µs, ms, or s), you need to use the following formula:

$$\text{Expect_time}(s) = \text{RELOAD} / (\text{SystemCoreClock-1}) \quad (4)$$

- If you need to start counting from the new RELOAD value immediately, you need to deinitialize the Timer first, and then initialize the Timer.

8.3.1.4 Registers

8.3.1.4.1 CTRL

- **Name**: Timer Control Register
- **Description**: The CTRL register enables the software to control the Timer.
- **Base Address**: 0x40000000 + x*0x1000
- **Offset**: 0x00
- **Reset Value**: 0x00

Note:

*x is used to identify the Timer Module. The “x” can be 0 or 1.

Table 8-44 CTRL Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3	INT_EN	RW	0x0	Timer interrupt enable **Value**: • 0x0: Disable the Timer interrupt. • 0x1: Enable the Timer interrupt.
2:1	RSVD	R		Reserved bits
0	EN	RW	0x0	Timer enable.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0: Disable the Timer. • 0x1: Enable the Timer.

8.3.1.4.2 VAL

- **Name**: Timer Value Register
- **Description**: The VAL register indicates the current value of the decrementing counter.
- **Base Address**: 0x40000000 + x*0x1000
- **Offset**: 0x04
- **Reset Value**: 0x01

Table 8-45 VAL Register

Bits	Field Name	RW	Reset	Description
31:0	CUR_VALUE	RW	0x1	Current value

8.3.1.4.3 RELOAD

- **Name**: Timer Reload Register
- **Description**: When this register is written to, the count value takes effect at the end of the current cycle.
- **Base Address**: 0x40000000 + x*0x1000
- **Offset**: 0x08
- **Reset Value**: 0x00

Table 8-46 RELOAD Register

Bits	Field Name	RW	Reset	Description
31:0	RELOAD_VALUE	RW	0x0	Sets reload value.

8.3.1.4.4 INT_STAT

- **Name**: Timer Interrupt Status Register
- **Description**: The INT_STAT register indicates the interrupt status from the counter. Writing one to the INT_STAT register clears the timer interrupt.
- **Base Address**: 0x40000000 + x*0x1000
- **Offset**: 0x0C
- **Reset Value**: 0x00000000

Table 8-47 INT_STAT Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	STAT	RW	0x0	Timer interrupt. Write 1 to clear.

8.3.2 Dual Timer

8.3.2.1 Introduction

The Dual Timer module consists of two programmable down counters (32-bit or 16-bit) that can generate interrupts when they reach 0. Since the Dual-Timer is in the MCU Subsystem domain, Dual-Timer interrupt is not available during sleep, and the running Dual-Timer will not prevent the MCU from sleeping. If the RELOAD register is configured with a value, the Dual-Timer immediately counts down from the RELOAD value; if the BG_LOAD register is configured with a value, the Dual-Timer starts counting down again with the BG_LOAD value after the current counting cycle is completed. If BG_LOAD value is set before Dual-Timer is initialized, the BG_LOAD value will be invalid after the Dual-Timer is initialized with RELOAD value.

8.3.2.2 Main Features

- A 32-bit or 16-bit counter
- One of the following timer modes:
  - Periodic
  - One-shot
  - Free-running
- The clock source of the Dual Timer follows the system clock
- Dual Timer has a prescaler that can divide down the enabled timer clock rate by 1, 16, or 256.
- Dual Timer pauses when the MCU is suspended.

8.3.2.3 Functional Description

In Dual Timer, both of the counters are defined by default. For each counter, the following operational modes are available:

- Free-running mode

  The counter wraps after reaching zero, and continues to count down from the maximum value (0xFFFFFFFF). This is the default mode.

- Periodic mode

  The counter generates an interrupt at a fixed interval, reloading the original value after wrapping past zeros.

- One-shot mode

  The counter generates an interrupt once. When the counter reaches 0, it halts until you reprogram it.

  When the value of the RELOAD register is configured to **0**, no count and no interrupt are generated.

The RELOAD value only represents the count-down number of timers. If you want to configure a specific duration (for example, µs, ms, or s), you need to use the following formula:

$$\text{Expect_time}(s) = \text{RELOAD} / (\text{SystemCoreClock} / \text{PRE} - 1) \quad (5)$$

where PRE stands for prescaled coefficient.

8.3.2.4 Registers

8.3.2.4.1 RELOAD

- **Name**: Dual Timer Reload Register
- **Description**: This register contains the value from which the counter is to decrease. The count is immediately restarted from the new RELOAD value.
- **Base Address**: 0x40002000
- **Offset**: 0x00 + x*0x20
- **Reset Value**: 0x00

Note:

*x is used to identify the counter of the Dual Timer Module. The “x” can be 0 or 1.

Table 8-48 RELOAD Register

Bits	Field Name	RW	Reset	Description
31:0	RELOAD_VALUE	RW	0x0	Reload value. Write to this register sets the current value.

8.3.2.4.2 VAL

- **Name**: Dual Timer Current Value Register
- **Description**: This register provides the current count value of the decrementing counter.
- **Base Address**: 0x40002000
- **Offset**: 0x04 + x*0x20
- **Reset Value**: 0xFFFFFFFF

Table 8-49 VAL Register

Bits	Field Name	RW	Reset	Description
31:0	CUR_VALUE	RO	0xFFFFFFFF	Current count value

8.3.2.4.3 CTRL

- **Name**: Dual Timer Control Register
- **Description**: This register enables the software to control the Dual Timer.
- **Base Address**: 0x40002000
- **Offset**: 0x08 + x*0x20
- **Reset Value**: 0x20

Table 8-50 CTRL Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7	EN	RW	0x0	Timer enable **Value**: • 0x0: Timer disabled; default • 0x1: Timer enabled
6	MODE	RW	0x0	Timer mode **Value**: • 0x0: Timer is in free-running mode; default • 0x1: Timer is in periodic mode
5	INT_EN	RW	0x1	Interrupt enable **Value**: • 0x0: Timer interrupt disabled • 0x1: Timer interrupt enabled; default
4	RSVD	R		Reserved bits
3:2	PRE	RW	0x0	Prescale bits **Value**: • 0x00: 0 stage of prescale, clock is divided by 1; default • 0x01: 4 stages of prescale, clock is divided by 16 • 0x10: 8 stages of prescale, clock is divided by 256 • 0x11: Undefined. Do not use this value.
1	SIZE	RW	0x0	Select 16-bit or 32-bit counter operation: **Value**: • 0x0: 16-bit counter; default • 0x1: 32-bit counter
0	COUNT_MODE	RW	0x0	Select wrap mode or one-shot mode: **Value**:

Bits	Field Name	RW	Reset	Description
				• 0x0: Wrap mode; default • 0x1: One-shot mode

8.3.2.4.4 INT_CLR

- **Name**: Dual Timer Interrupt Clear Register
- **Description**: Any write to the INT_CLR register clears the interrupt output from the counter.
- **Base Address**: 0x40002000
- **Offset**: 0x0C + x*0x20
- **Reset Value**: 0x00

Table 8-51 INT_CLR Register

Bits	Field Name	RW	Reset	Description
31:0	STAT	WO	0x0	Clear the Dual-Timer interrupt event.

8.3.2.4.5 RAW_INT_STAT

- **Name**: Dual Timer Raw Interrupt Status Register
- **Description**: The RAW_INT_STAT register indicates the raw interrupt status from the counter.
- **Base Address**: 0x40002000
- **Offset**: 0x10 + x*0x20
- **Reset Value**: 0x00

Table 8-52 RAW_INT_STAT Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	STAT	RO	0x0	Raw interrupt status from the counter

8.3.2.4.6 INT_STAT

- **Name**: Dual-Timer interrupt enable status register
- **Description**: The INT_STAT register indicates the masked interrupt status from the counter.
- **Base Address**: 0x40002000
- **Offset**: 0x14 + x*0x20
- **Reset Value**: 0x00

Table 8-53 INT_STAT Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	STAT	RO	0x0	Enable interrupt status from the counter.

8.3.2.4.7 BG_LOAD

- **Name**: Dual-Timer background reload register
- **Description**: When the BG_LOAD value is set, the counter will count down again from the BG_LOAD value after completing the current counting cycle.
- **Base Address**: 0x40002000
- **Offset**: 0x18 + x*0x20
- **Reset Value**: 0x00

Table 8-54 BG_LOAD Register

Bits	Field Name	RW	Reset	Description
31:0	BG_LOAD_VALUE	RW	0x0	The BG_LOAD does not cause the counter to immediately restart from the new value.

8.3.3 PWM

8.3.3.1 Introduction

GR5526 SoC has two instances of Pulse Width Modulation (PWM) module. The PWM generates successive pulses with variable duty cycles that can be converted to analog levels by external discrete components, such as controlling LED brightness via a PWM signal.

8.3.3.2 Main Features

- Configurable output frequency and dynamic frequency configuration
- The three PWM channel outputs (pwma, pwmb, and pwmc) can only be configured independently in the flicker mode.
- Two operation modes: flicker mode and breath mode
- Two alignment modes: edge alignment mode and center alignment mode
- Configurable duty cycles in flicker mode
- Three configuration modes: synchronize-all, synchronize-separate, and asynchronous
- Polarity switching to support positive-drive and negative-drive mode
- Optional stop I/O voltage level in active/idle mode: low level or high level

8.3.3.3 Block Diagram

The PWM block diagram is illustrated in Figure 8-2.

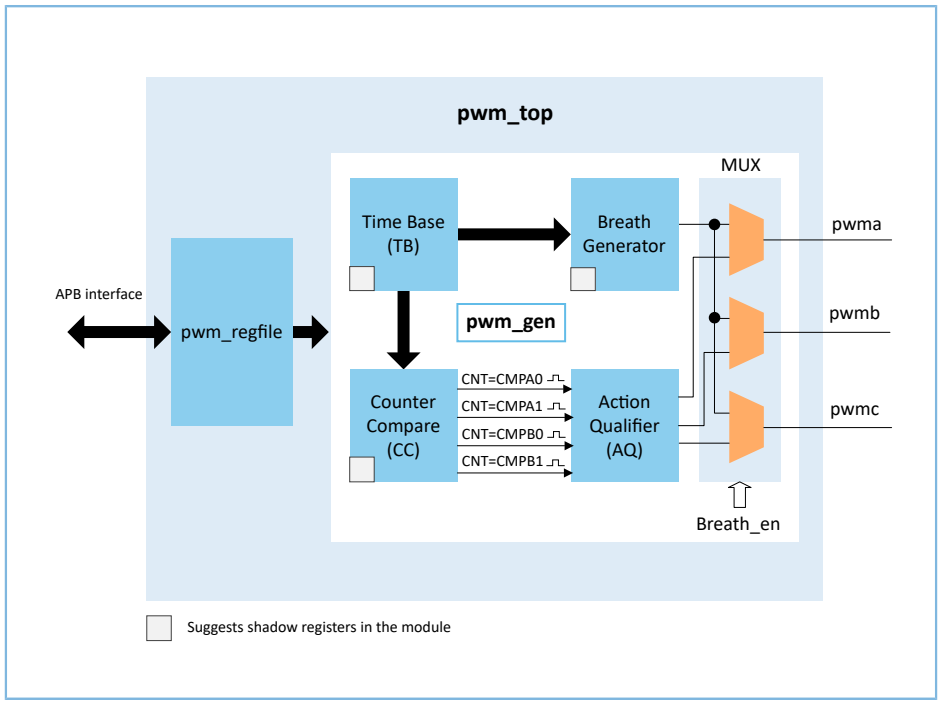

Figure 8-2 PWM block diagram

8.3.3.4 Functional Description

The PWM is capable of configuring output frequency dynamically.

The PWM block can provide three independent channel outputs: pwma, pwmb and pwmc in flicker mode.

The PWM block has two operation modes: flicker mode and breath mode.

In flicker mode, the PWM module outputs successive pulses with a certain frequency and duty as configured. In breath mode, the duty of PWM output periodically changes from 0% to 100%, and then 100% to 0% uniformly. The change period is configurable.

In flicker mode, the duty of PWM can be configured in the range of 0% to 100%. The three outputs can be configured independently, and they share one output frequency.

In flicker mode, both edge alignment and center alignment are supported.

Choose edge alignment mode, as shown in the Figure 8-3. All open channels are aligned at the beginning of each duty cycle.

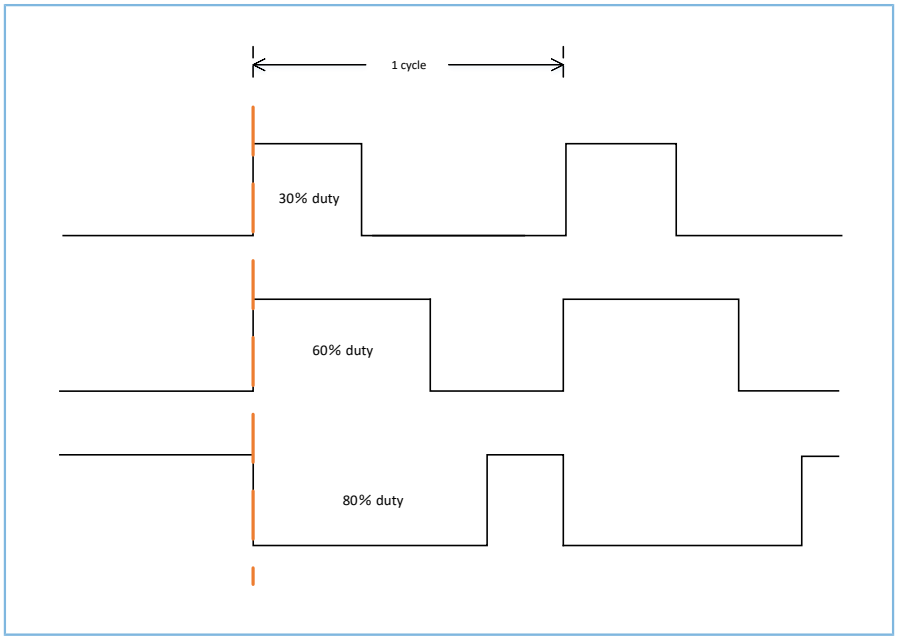

Figure 8-3 PWM edge alignment

Choose center alignment mode, as shown in the Figure 8-4. All open channels are aligned in the middle of each duty cycle.

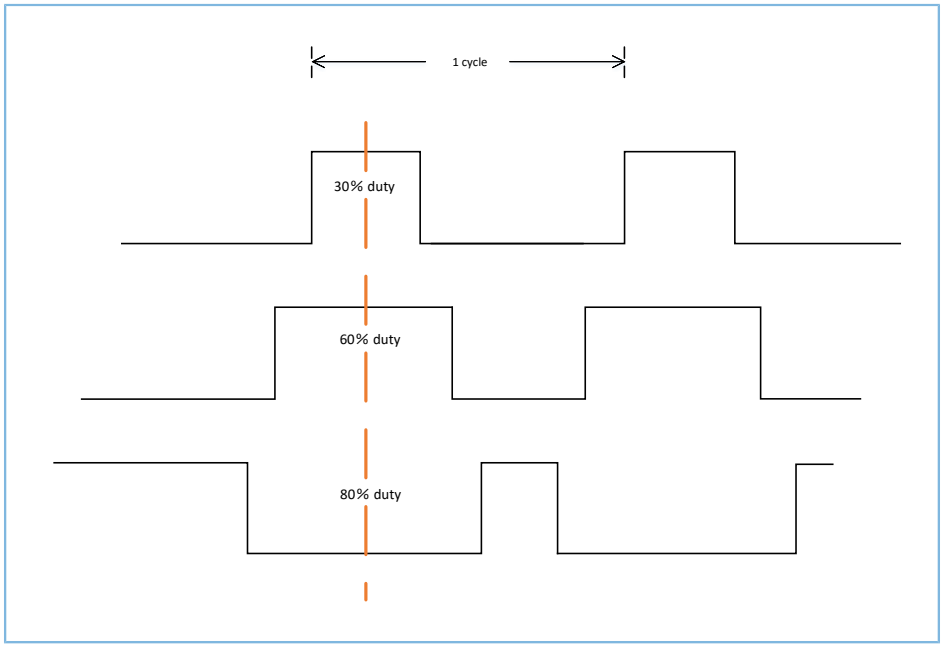

Figure 8-4 PWM center alignment

The configured data can be updated into active registers in three modes: synchronize-all mode (enabled by UPDATE_SYNC_AE), synchronize-separate mode (enabled by UPDATE_SYNC_Sxx) and asynchronous update mode (enabled by UPDATE_SYNC_AE and UPDATE_SYNC_Sxx).

In breath mode, the breathing cycle can be set by BR_PRD, which represents the number of PWM cycles required for the duty cycle 0%→100% and then 100%→0%

In breath mode, a configurable duration of breath hold state (HOLD) is supported. The breath hold state is set between two adjacent breath processes, which are duty change (0%→100%→0%), breath hold, and duty change (0%→100%→0%). In the hold state, the PWM output is driven by inactive state.

Both positive-drive mode and negative-drive mode are supported and can be configured in PD_x_EN. In positive-drive mode, a valid output is configured by setting the PWM output logic level to **1**, and an invalid output is set when the PWM output logic level is **0**. In negative-drive mode, a valid output is configured by setting the PWM output logic level to **0**, and an invalid output is set when PWM output logic level is 1.

In flicker mode, if the stop level is configured (the default value of FLICKER_PAUSE_LEVEL_x is **0** when not set), when PWM stops, the I/O level of the corresponding channel will be pulled to the set value. In breath mode, the I/O levels of all channels are the same and set by BREATH_PAUSE_LEVEL when PWM stops. When the MCU enters sleep state, the stop level remains high in both flicker mode and breath mode.

8.3.3.4.1 Operation

There are several shadow registers in the PWM module. The write operation will first change the shadow registers, and then will update the value of active registers from the shadow registers under certain conditions.

Active registers update conditions:

1. In synchronize-all update mode (**UPDATE_SYNC_AE** = **0x1**), all effective registers of flicker mode or breath mode will be updated simultaneously. In flicker mode **(BREATH_EN** = **0**), after writing to PRD, CMPxx and AQ_CTRL will update from shadow registers simultaneously when the Time Base counter counts to a value of PRD minus 1.

Figure 8-5 Flicker mode waveform

In breath mode (BREATH_EN = **1**), after writing to PRD, BR_PRD, and HOLD will update from shadow registers simultaneously when the current breath process ends (the moment PWM duty reduces to 0%). When using the app_pwm_update_freq interface, the PWM frequency will be updated immediately.

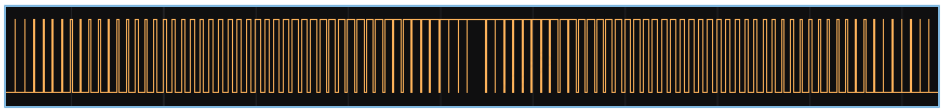

Figure 8-6 Breath mode waveform

2. In synchronize-separate update mode (**UPDATE_SYNC_AE** = **0x0**), PRD, CMPxx, and AQ_CTRL will update when the Time Base counter counts to value of the PRD register minus 1 if the corresponding UPDATE_SYNC_Sxx bit is set. BR_PRD and HOLD will update when the current breath process ends and the corresponding UPDATE_SYNC_Sxx bit is set.

3. In asynchronous update mode (**UPDATE_SYNC_AE** = **0x0** && **UPDATE_SYNC_Sxx** = **0x0**), all registers will update instantly.

In initial state, set the update mode first, then PRD, CMPxx, AQ_CTRL, BR_PRD, and HOLD registers are configured, and then, drive mode is configured to choose the positive-drive mode or negative-drive mode, finally, choose breath mode or flicker mode and enable PWM.

8.3.3.4.2 Duty Cycle Calculation

The PWM duty cycle can be configured as an integer from 0 to 100, depending on the following calculation formulas of the duty cycle accuracy "n".

- In PWM_ALIGNED_EDGE mode:

$$n = f_{PWM} \div f_{SYSCLK} \times 100 \tag{6}$$

- In PWM_CENTER_EDGE mode:

$$n = f_{PWM} \div f_{SYSCLK} \times 200 \tag{7}$$

For example, when the clock frequency is 64 MHz, and the duty cycle is in the range of 0 to 100, the maximum PWM frequency can only be set to 640 kHz in the PWM_ALIGNED_EDGE mode and 320 kHz in the PWM_ALIGNED_CENTER mode.

8.3.3.4.3 Recommended Setting in Breath Mode

In breath mode, in order to satisfy that the duty cycle of the waveform changes from 0 to 100 and then from 100 to 0, it is allowed to set the values of PRD and BR_PRD to an integer that is multiple(s) of 128. The freq, bperiod and hperiod in the structure pwm_init_t need to meet the following formula during initialization:

$$freq \times bperiod = 1000 \times k\,(k \geq 2 \&\& k \in interger) \tag{8}$$

$$\frac{SYS_{CLK}}{1000} \times \text{hperiod} \leq 16777215 \tag{9}$$

where, k represents the number of PWM cycles required for the duty cycle from 0% to 100%; SYS_{CLK} represents the system clock frequency.

8.3.3.5 Registers

8.3.3.5.1 MODE

- **Name**: Mode Register
- **Description**: This register sets the PWM operation mode.
- **Base Address**: 0x4000CB00 + x*0x100
- **Offset**: 0x0
- **Reset Value**: 0x00000000

Note:

*x is used to identify the PWM Module instance. The “x” can be 0 or 1.

Table 8-55 Mode Register

Bits	Field Name	RW	Reset	Description
31:10	RSVD	R		Reserved bits
9	BREATH_PAUSE_LEVEL	RW	0x0	PWMA/B/C output IO level when suspending in Breath Mode
8	FLICKER_PAUSE_LEVEL_C	RW	0x0	PWMC output IO level when suspending in Flicker Mode
7	FLICKER_PAUSE_LEVEL_B	RW	0x0	PWMB output IO level when suspending in Flicker Mode
6	FLICKER_PAUSE_LEVEL_A	RW	0x0	PWMA output IO level when suspending in Flicker Mode
5	PD_C_EN	RW	0x0	PWMC positive-drive mode enable. **Value**: • 0x0: negative-drive mode • 0x1: positive-drive mode
4	PD_B_EN	RW	0x0	PWMB positive-drive mode enable. **Value**: • 0x0: negative-drive mode • 0x1: positive-drive mode
3	PD_A_EN	RW	0x0	PWMA positive-drive mode enable. **Value**: • 0x0: negative-drive mode • 0x1: positive-drive mode
2	BREATH_EN	RW	0x0	Breath mode enable. **Value**: • 0x0: flicker mode • 0x1: breath mode
1	PAUSE	RW	0x0	PWM pause signal. **Value**: • 0x0: Ongoing • 0x1: Pause
0	ENABLE	RW	0x0	Enable PWM. **Value**: • 0x0: Disable • 0x1: Enable

8.3.3.5.2 UPDATE

- **Name**: Update Register

- **Description**: This register is used to set the synchronous update mode.
- **Base Address**: 0x4000CB00 + x*0x100
- **Offset**: 0x4
- **Reset Value**: 0x00000000

Table 8-56 Update Register

Bits	Field Name	RW	Reset	Description
31:18	RSVD	R		Reserved bits
17	UPDATE_SYNC_SAQCTRL	RW	0x0	Separate update enable of AQ_CTRL **Value**: • 0x0: Disable • 0x1: Enable
16	UPDATE_SYNC_SHOLD	RW	0x0	Separate update enable of HOLD **Value**: • 0x0: Disable • 0x1: Enable
15	UPDATE_SYNC_SBRPRD	RW	0x0	Separate update enable of BR_PRD **Value**: • 0x0: Disable • 0x1: Enable
14	UPDATE_SYNC_SCMPC1	RW	0x0	Separate update enable of CMPC1 **Value**: • 0x0: Disable • 0x1: Enable
13	UPDATE_SYNC_SCMPC0	RW	0x0	Separate update enable of CMPC0 **Value**: • 0x0: Disable • 0x1: Enable
12	UPDATE_SYNC_SCMPB1	RW	0x0	Separate update enable of CMPB1 **Value**: • 0x0: Disable • 0x1: Enable
11	UPDATE_SYNC_SCMPB0	RW	0x0	Separate update enable of CMPB0 **Value**: • 0x0: Disable

Bits	Field Name	RW	Reset	Description
				• 0x1: Enable
10	UPDATE_SYNC_SCMPA1	RW	0x0	Separate update enable of CMPA1 **Value**: • 0x0: Disable • 0x1: Enable
9	UPDATE_SYNC_SCMPA0	RW	0x0	Separate update enable of CMPA0 **Value**: • 0x0: Disable • 0x1: Enable
8	UPDATE_SYNC_SPRD	RW	0x0	Separate update enable of PRD **Value**: • 0x0: Disable • 0x1: Enable
7:2	RSVD	R		Reserved bits
1	UPDATE_SYNC_AE	RW	0x0	Synchronous update enable **Value**: • 0x0: Disable • 0x1: Enable
0	UPDATE_SYNC_AG	R	0x0	Synchronous update ongoing **Value**: • 0x0: Not ongoing • 0x1: Ongoing

8.3.3.5.3 PRD

- **Name**: Period Register
- **Description**: This register is used to set the PWM cycle.
- **Base Address**: 0x4000CB00 + x*0x100
- **Offset**: 0x8
- **Reset Value**: 0x64

Table 8-57 Period Register

Bits	Field Name	RW	Reset	Description
31:0	PRD	RW	0x64	The period of PWM output, PRD=f_{CLK}/f_{PWM}

8.3.3.5.4 CMPA0

- **Name**: Compare A0 Register
- **Description**: This register controls the PWM duty of Compare register 0 of channel A.
- **Base Address**: 0x4000CB00 + x*0x100
- **Offset**: 0xC
- **Reset Value**: 0x00000000

Table 8-58 Compare A0 Register

Bits	Field Name	RW	Reset	Description
31:0	CMPA0	RW	0x0	PWMA duty control register0

8.3.3.5.5 CMPA1

- **Name**: Compare A1 Register
- **Description**: This register controls the PWM duty of Compare register 1 of channel A.
- **Base Address**: 0x4000CB00 + x*0x100
- **Offset**: 0x10
- **Reset Value**: 0x00000000

Table 8-59 Compare A1 Register

Bits	Field Name	RW	Reset	Description
31:0	CMPA1	RW	0x0	PWMA duty control register1

8.3.3.5.6 CMPB0

- **Name**: Compare B0 Register
- **Description**: This register controls the PWM duty of Compare register 0 of channel B.
- **Base Address**: 0x4000CB00 + x*0x100
- **Offset**: 0x14
- **Reset Value**: 0x00000000

Table 8-60 Compare B0 Register

Bits	Field Name	RW	Reset	Description
31:0	CMPB0	RW	0x0	PWMB duty control register0

8.3.3.5.7 CMPB1

- **Name**: Compare B1 Register
- **Description**: This register controls the PWM duty of Compare register 1 of channel B.
- **Base Address**: 0x4000CB00 + x*0x100
- **Offset**: 0x18
- **Reset Value**: 0x00000000

Table 8-61 Compare B1 Register

Bits	Field Name	RW	Reset	Description
31:0	CMPB1	RW	0x0	PWMB duty control register1

8.3.3.5.8 CMPC0

- **Name**: Compare C0 Register
- **Description**: This register controls the PWM duty of Compare register 0 of channel C.
- **Base Address**: 0x4000CB00 + x*0x100
- **Offset**: 0x1C
- **Reset Value**: 0x00000000

Table 8-62 Compare C0 Register

Bits	Field Name	RW	Reset	Description
31:0	CMPC0	RW	0x0	PWMC duty control register0

8.3.3.5.9 CMPC1

- **Name**: Compare C1 Register
- **Description**: This register controls the PWM duty of Compare register 1 of channel C.
- **Base Address**: 0x4000CB00 + x*0x100
- **Offset**: 0x20
- **Reset Value**: 0x00000000

Table 8-63 Compare C1 Register

Bits	Field Name	RW	Reset	Description
31:0	CMPC1	RW	0x0	PWMC duty control register1

8.3.3.5.10 AQ_CTRL

- **Name**: Action Qualifier Control Register

- **Description**: This register is used to set actions when counter reaches the compare value.
- **Base Address**: 0x4000CB00 + x*0x100
- **Offset**: 0x24
- **Reset Value**: 0x00000000

Table 8-64 Action Qualifier Control Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R		Reserved bits
11:10	AQ_CTRL_C1	RW	0x0	Action of event CNT_CMPC1 control register **Value**: • 0x0: Do nothing • 0x1: Clear • 0x2: Set • 0x3: Toggle
9:8	AQ_CTRL_C0	RW	0x0	Action of event CNT_CMPC0 control register **Value**: • 0x0: Do nothing • 0x1: Clear • 0x2: Set • 0x3: Toggle
7:6	AQ_CTRL_B1	RW	0x0	Action of event CNT_CMPB1 control register **Value**: • 0x0: Do nothing • 0x1: Clear • 0x2: Set • 0x3: Toggle
5:4	AQ_CTRL_B0	RW	0x0	Action of event CNT_CMPB0 control register **Value**: • 0x0: Do nothing • 0x1: Clear • 0x2: Set • 0x3: Toggle
3:2	AQ_CTRL_A1	RW	0x0	Action of event CNT_CMPA1 control register **Value**: • 0x0: Do nothing

Bits	Field Name	RW	Reset	Description
				• 0x1: Clear • 0x2: Set • 0x3: Toggle
1:0	AQ_CTRL_A0	RW	0x0	Action of event CNT_CMPA0 control register **Value**: • 0x0: Do nothing • 0x1: Clear • 0x2: Set • 0x3: Toggle

8.3.3.5.11 BR_PRD

- **Name**: Breath Period Register
- **Description**: This register configures the period of breath mode.
- **Base Address**: 0x4000CB00 + x*0x100
- **Offset**: 0x28
- **Reset Value**: 0x00000000

Table 8-65 Breath Period Register

Bits	Field Name	RW	Reset	Description
31:0	BREATH_PRD	RW	0x0	Register for breath period that is the required time (number of clock) that the duty changes from 0% to 100% in breath mode.

8.3.3.5.12 HOLD

- **Name**: Hold Register
- **Description**: This register configures the hold period in breath mode.
- **Base Address**: 0x4000CB00 + x*0x100
- **Offset**: 0x2C
- **Reset Value**: 0x00000000

Table 8-66 Hold Register

Bits	Field Name	RW	Reset	Description
31:24	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
23:0	HOLD	RW	0x0	Breath hold control register. The value should be the required number of clock in breath hold state.

8.4 Communication Interfaces

8.4.1 UART

8.4.1.1 Introduction

GR5526 SoC has 6 instances of Universal Asynchronous Receiver/Transmitter (UART) module. The UART complies with the industry-standard 16550. All the six UART instances support full duplex asynchronous serial communication with hardware flow control. The register address space is relocated to 32-bit data boundaries for Advance Peripheral Bus (APB) implementation.

8.4.1.2 Main Features

- Full-duplex asynchronous communications
- Programmable character properties
- Transmit and receive FIFO depth at 128 bytes
- Hardware flow control support
- DMA support
- IrDA 1.0 SIR support
- 6 UART instances

8.4.1.3 Block Diagram

The functional block diagram of UART is shown in Figure 8-7.

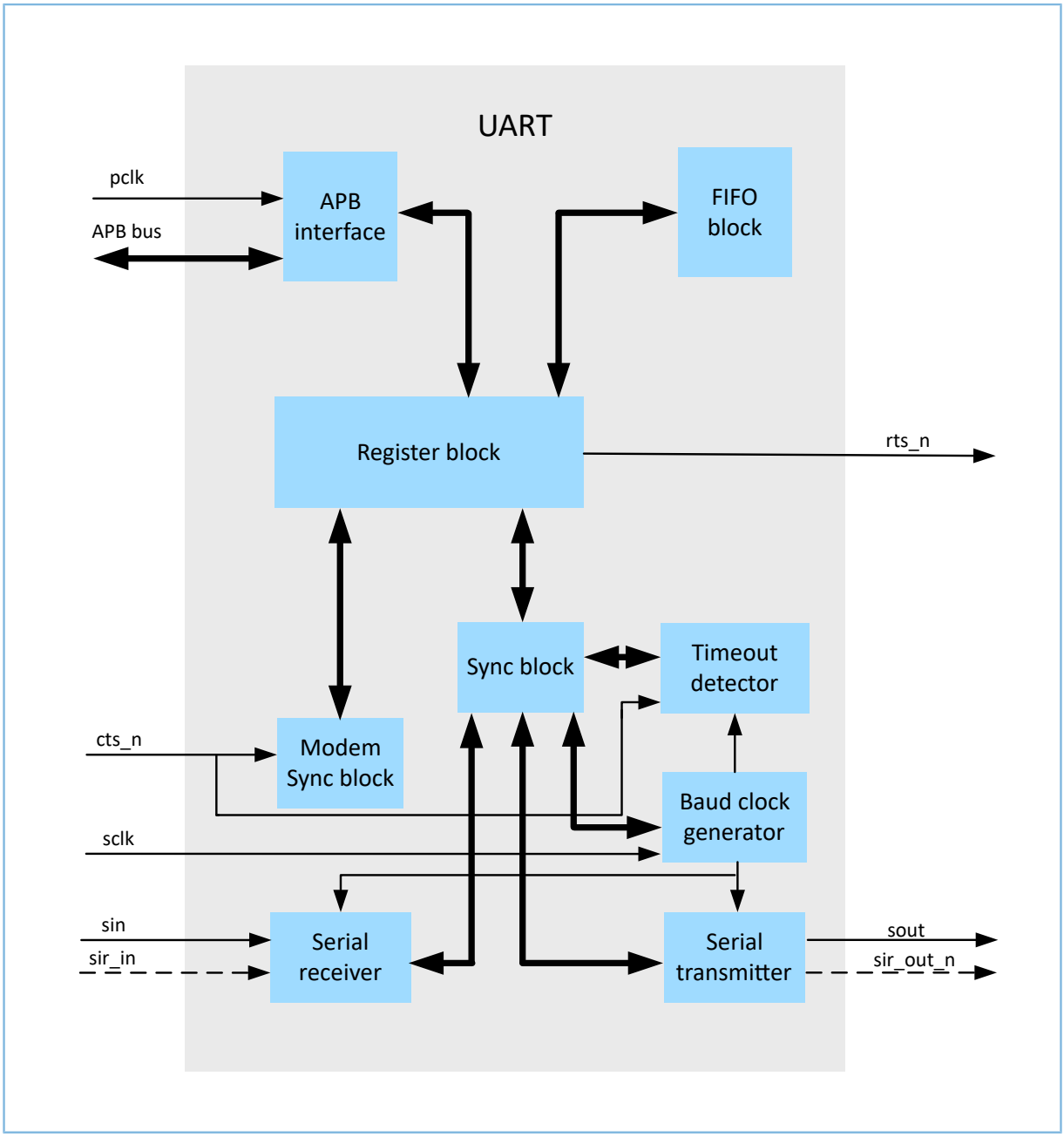

Figure 8-7 UART block diagram

The following list describes each of the major blocks shown in Figure 8-7.

- Reset Block resets the clock domain.
- APB Interface connects to APB bus.
- Register Block is responsible for the main UART functionality including control, status, and interrupt generation.
- Modem Sync Block synchronizes the modem input signal.
- FIFO Block is responsible for FIFO control and storage — when using internal RAM — or optionally signaling to control external RAM.

- Sync Block is implemented when the peripheral is configured to have a separate serial data clock.
- Timeout Detector Block indicates the absence of character data movement in the receiver FIFO within a given time period; this is used to generate character timeout interrupts when enabled.
- Baud Clock Generator produces the transmitter and receiver baud clock along with the output reference clock signal.
- Serial Transmitter converts the parallel data — written to the UART — into serial form and adds all additional bits, as specified by the control register, for transmission.
    - Serial UART
    - IrDA 1.0 SIR
- Serial Receiver converts the serial data character — specified by the control register — received in UART form. This block controls:
    - Parity error detection
    - Framing error detection
    - Line break detection

8.4.1.4 Functional Description

8.4.1.4.1 UART (RS232) Serial Protocol

The serial communication between the UART and a selected device is asynchronous, so additional start and stop bits are added to the serial data to indicate the beginning and end. Utilizing these bits allows two devices to be synchronized. This structure of serial data, accompanied by start and stop bits, is referred to as a character, as shown in Figure 8-8.

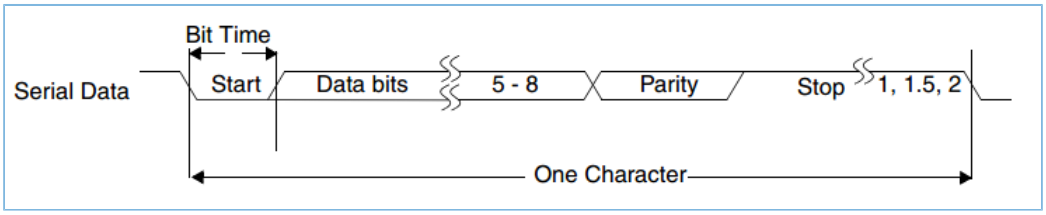

Figure 8-8 Timing diagram

An additional parity bit can be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure in order to provide the UART with the ability to perform simple error checking on the received data.

The UART Line Control Register (LCR) is used to control the serial character characteristics. The individual bits of the data word are sent after the start bit, starting with the least-significant bit (LSB). These are followed by the optional parity bit, followed by the stop bit(s), which can be 1, 1.5, or 2.

All the bits in the transmission are transmitted for exactly the same time duration; the exception to this is the half-stop bit when 1.5 stop bits are used. This duration is referred to as a Bit Period or Bit Time; one Bit Time equals sixteen baud clocks.

To ensure stability on the line, the receiver samples the serial input data at approximately the midpoint of the Bit Time once the start bit has been detected. The exact number of baud clocks that are transmitted by each bit is known, so calculating the midpoint for sampling is not difficult; the midpoint for sampling occurs every sixteen baud clocks after sampling is performed at the midpoint of the start bit.

Together with serial input debouncing, this sampling helps to avoid the detection of false start bits. Short glitches are filtered out by debouncing, and no transition is detected on the line. If a glitch is wide enough to avoid filtering by debouncing, a falling edge is detected. However, a start bit is detected only if the line is again sampled low after half a Bit Time has elapsed. Figure 8-9 shows the sampling points of the first two bits in a serial character.

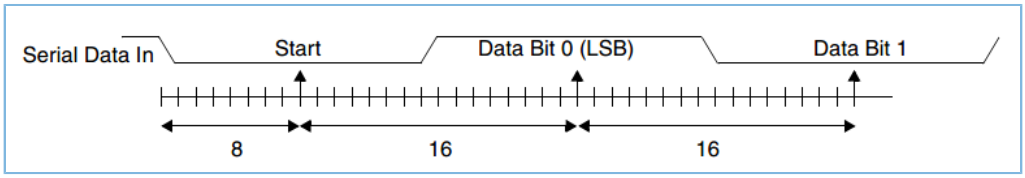

Figure 8-9 Receiver serial data sample points

As part of the 16550 standard, an optional baud clock reference output signal (baudout_n) provides timing information for receiving devices that require it. The baud rate of the UART is controlled by the serial clock – sclk (in asynchronous serial clock implementation; see Figure 8-7)– and the Divisor Latch Register (DLH and DLL).

Serial data baud rate can be calculated as follows: Baud rate = (System clock frequency)/(16*Divisor). The Divisor is composed of an integer part (DIV_LATCH_HIGH/DIV_LATCH_LOW register) and a fractional part (4-bit DIV_LATCH_FRACTION register).

8.4.1.4.2 Fractional Baud Rate Support

UART supports fractional baud rate that enables users to program the fractional part of the divisor value to generate a fractional baud rate that results in reduced frequency error. The baud rate of UART is controlled by sclk and the Divisor Latch Register (DLH and DLL).

The baud rate is determined by the following factors:

- Serial clock operating frequency (sclk in asynchronous serial clock implementation)
- The desired baud rate, or Baud rate (selected)
- The baud rate generator divisor value, DIVISOR (composed of DLH & DLL registers).
- The acceptable baud-rate error: Percentage ERROR.

The equation to calculate the baud rate is as follows:

$$\text{Baud Rate} = \frac{\text{Serial Clock Operating Frequency}}{(16 \times \text{DIVISOR})} \tag{10}$$

where

DIVISOR – Number (in hexadecimal) to program the DLL and DLH. Serial clock operating frequency – Frequency at sclk pin of UART. The Error between the Baud rate and Baud rate (selected) is given as:

$$Percentage\ \ Error = \frac{|BaudRate - BaudRate\ \ (Selected)|}{BaudRate} \times 100 \tag{11}$$

When configuring the UART for Fractional Baud Rate, pay attention to the following:

- The configurable parameter DLF_SIZE is used to choose the width of the register that stores fractional part of the divisor.
- The fractional value of the divisor is programmed in the Divisor Latch Fraction (DLF) register. The fractional value is computed by using the (Divisor Fraction Value) / (2^DLF_SIZE) formula. Table 8-67 shows fractional values when the **DLF_SIZE** = **4**.

Table 8-67 Divisor latch fractional values

DLF Value	Fraction	Fractional Value
0000	0/16	0.0000
0001	1/16	0.0625
0010	2/16	0.125
0011	3/16	0.1875
0100	4/16	0.25
0101	5/16	0.3125
0110	6/16	0.375
0111	7/16	0.4375
1000	8/16	0.5
1001	9/16	0.5625
1010	10/16	0.625
1011	11/16	0.6875
1100	12/16	0.75
1101	13/16	0.8125
1110	14/16	0.875
1111	15/16	0.9375

- The programmable fractional baud rate divisor enables a finer resolution of baud clock than the conventional integer divider. The programmable fractional baud clock divider allows for the programmability of both an integer divisor as well as fractional component. The average frequency of the baud clock from the fractional baud rate divisor is dependent upon both the integer divisor and the fractional component, thereby providing a finer resolution to the average frequency of the baud clock.

$$\text{Baud Rate Divisor} = \frac{\text{Serial Clock Frequency}}{(16 \times \text{Required Baud Rate})} = BRD_I + BRD_F \quad (12)$$

where

BRD_I: Integer part of the divisor

BRD_F: Fractional part of the divisor

8.4.1.4.3 FIFO Support

The UART module provides two 128-byte FIFO submodules: one (UART TX FIFO) for transmission and one (UART RX FIFO) for reception. By configuring the Shadow RCVR Trigger (SRT) register, you can program the trigger level in the RX FIFO at which the Received Data Available Interrupt is generated. Available configurations include 1 character, 1/4 full, 1/2 full, and 2 less than full. For example, if the 1/4 full option is selected for the receiver FIFO, the UART generates a receive interrupt after 32 data bytes are received. By configuring the Shadow TX Empty Trigger (STET) register, you can program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt is generated. Available configurations include empty, 2 characters, 1/4 full, and 1/2 full. By configuring the Shadow FIFO Enable (SFE) register, you can enable/disable transmit and receive FIFOs.

8.4.1.4.4 Clock Support

UART is configured to have two system clocks (pclk and sclk). Pclk is the APB clock and sclk is the serial clock (See Figure 8-7). This mode accommodates accurate serial baud rate settings, as well as APB bus interface requirements. In dual clock mode, a synchronization module is implemented for synchronization of all control and data across the two-system clock boundaries. There are two types of signal synchronization:

- Data-synchronized signals – full synchronization handshake takes place on signals.
- Level-synchronized signals – signals are passed through two destination clock registers.

Both synchronization types incur additional data path latencies. However, this additional latency has no negative effect on received or transmitted data, other than to limit how faster sclk can be in relation to pclk for serial communications with no idle assertion.

A serial clock that exceeds this limit does not leave enough time for a complete incoming character to be received and pushed into the receiver FIFO. To ensure that you do not exceed the limit, the following equation must hold true:

$$((2 * \text{pclk_cycles}) + 4) < (39 * (\text{Baud Divisor})) \quad (13)$$

Where: pclk_cycles is expressed in sclk cycles, and Baud Divisor is determined by DLH and DLL. In most cases, the pclk signal is faster than sclk, so this should not be an issue.

8.4.1.4.5 Interrupts

Assertion of the UART interrupt output signal occurs whenever one of the several prioritized interrupt types are enabled and active. When an interrupt occurs, the master accesses the IIR register. The following interrupt types can be enabled with the IER register:

- Receiver Error
- Receiver Data Available

- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE interrupt mode)
- Modem Status

These interrupt types are explained in detail in Table 8-68.

Table 8-68 Interrupt control functions

Interrupt ID	**Interrupt Set and Reset Functions**			
Bit3 Bit2 Bit1 Bit0	**Priority Level**	**Interrupt Type**	**Interrupt Source**	**Interrupt Reset Control**
0 0 0 1	-	None	None	-
0 1 1 0	Highest	Receiver line status	Overrun/Parity/Framing errors, break interrupt, or address received interrupt	For overrun/parity/framing/break interrupt reset control, the behavior is as follows: • If LSR_STATUS_CLEAR=0 (RBR Read or LSR Read), the status is cleared on: Reading the line status register. In addition to an LSR read, the Receiver line status is also cleared when RX_FIFO is read. • If LSR_STATUS_CLEAR=1 (LSR Read), the status is cleared only on: Reading the line status register. For address received interrupt, the status is cleared on: Reading the LINE_STAT register.
0 1 0 0	Second	Received data available	Receiver data available (non-FIFO mode or FIFOs disabled) or RCVR FIFO trigger level reached (FIFO mode and FIFOs enabled)	Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO increases above the trigger level (FIFO mode and FIFOs enabled).
1 1 0 0	Second	Character time indication	No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1 character in it during this time	Reading the receiver buffer register
0 0 1 0	Third	Transmit holding register empty	Transmitter holding register empty (Prog. THRE Mode disabled) or XMIT FIFO at or below threshold (Prog. THRE Mode enabled)	Reading the IIR register (if source of interrupt); or, writing into THR (FIFOs or THRE Mode not selected or disabled) or XMIT FIFO above threshold (FIFOs and THRE Mode selected and enabled)

Interrupt ID	Interrupt Set and Reset Functions			
Bit3 Bit2 Bit1 Bit0	**Priority Level**	**Interrupt Type**	**Interrupt Source**	**Interrupt Reset Control**
0 0 0 0	Fourth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect. Note: If auto flow control mode is enabled, a change in Clear to Send (CTS) (that is, DCTS set) does not cause an interrupt.	Reading the Modem status register

8.4.1.4.6 Auto Flow Control

The UART can be configured to have a 16750-compatible Auto Request to Send (RTS) and Auto Clear To Send (CTS) serial data flow control mode available; if FIFOs are not implemented, this mode cannot be selected. When Auto Flow Control is not selected, none of the corresponding logics is implemented, and the mode cannot be enabled, reducing overall gate counts. When Auto Flow Control mode is selected, it can be enabled with the Modem Control Register (MCR[5]). Figure 8-10 shows a block diagram of the Auto Flow Control functionality.

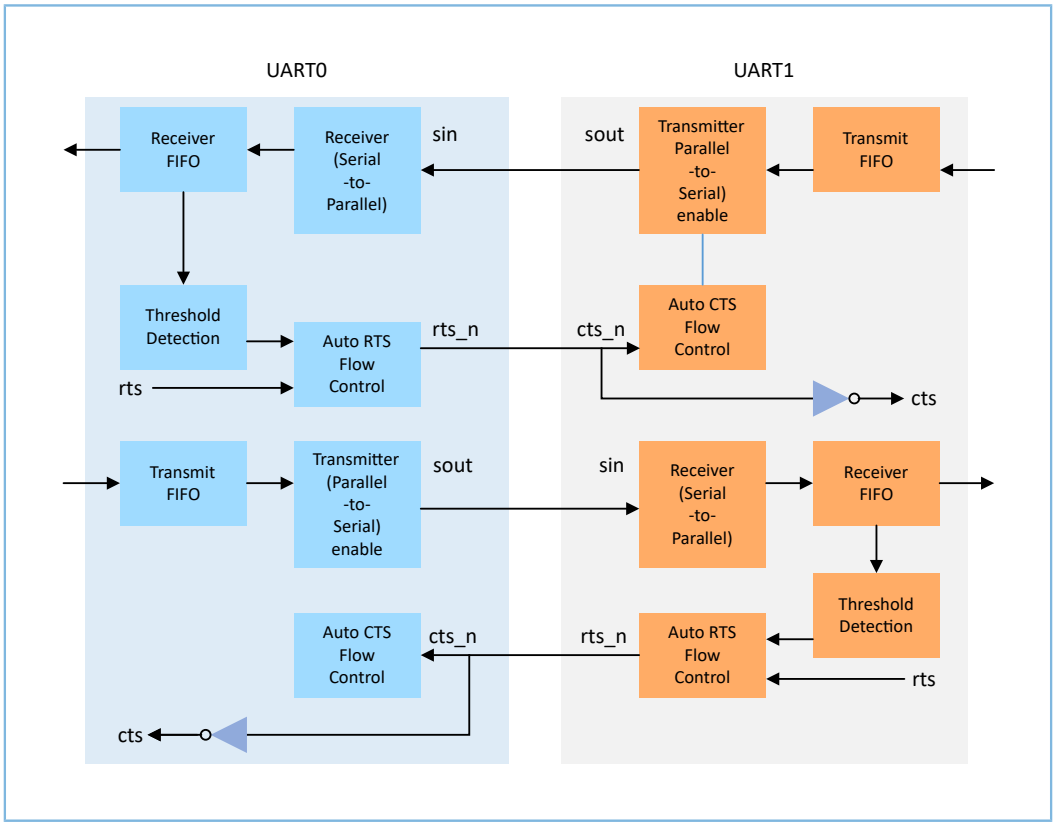

Figure 8-10 Auto Flow Control Block Diagram

- Auto RTS becomes active when all the following conditions occur:
  - Auto Flow Control is selected during configuration
  - FIFOs are implemented
  - RTS (MCR[1] bit and MCR[5] bit are both set)
  - FIFOs are enabled (FCR[0]) bit is set)
  - SIR mode is disabled (MCR[6] bit is not set)

    When Auto RTS is enabled, the rts_n output is forced inactive (high) when the FIFO is almost full, where “almost full” refers to two available slots in the FIFO. When rts_n is connected to the cts_n input of another UART device, the other UART stops sending serial data until the receiver FIFO has available space; that is, until it is completely empty.

    The selectable receiver FIFO threshold values are:

    - 1
    - 1/4
    - 1/2
    - 2 less than full

  Since one additional character can be transmitted to the UART after rts_n has become inactive—due to data already having entered the transmitter block in the other UART— setting the threshold to “2 less than full” allows maximum use of the FIFO with a safety zone of one character.

  When the receiver FIFO becomes completely empty by reading the Receiver Buffer Register (RBR), the rts_n again becomes active (low), signaling the other UART to continue sending data. Figure 8-11 shows a timing diagram of the Auto RTS operation.

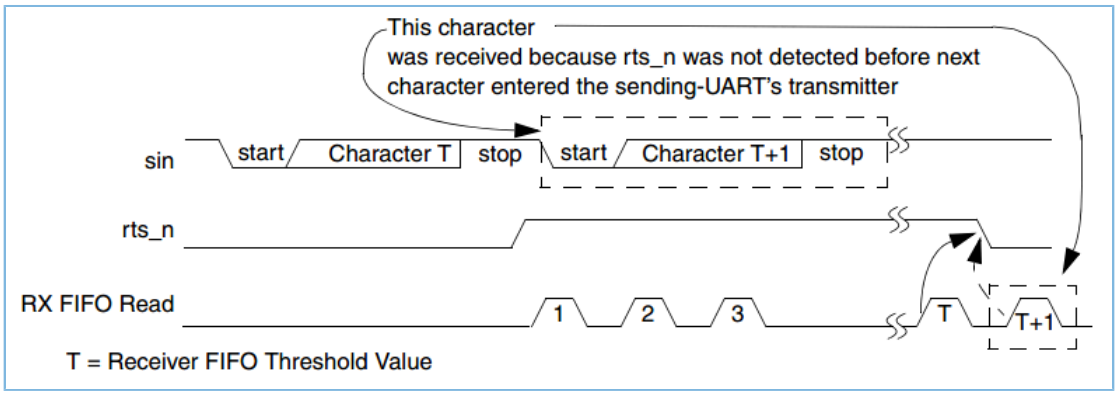

Figure 8-11 Auto RTS timing

- Auto CTS becomes active when all the following conditions occur:
  - Auto Flow Control is selected during configuration
  - FIFOs are implemented
  - AFCE (MCR[5] bit = **1**)

- FIFOs are enabled through FIFO Control Register FCR[0] bit
- SIR mode is disabled (MCR[6] bit = 0)

  When Auto CTS is enabled (active), the UART transmitter is disabled whenever the cts_n input becomes inactive (high); this prevents overflowing the FIFO of the receiving UART. If the cts_n input is not inactivated before the middle of the last stop bit, another character is transmitted before the transmitter is disabled. While the transmitter is disabled, the transmitter FIFO can still be written to, and even overflowed. Therefore, when using this Auto CTS mode, the following happens:

- UART status register can be read to check if transmit FIFO is full (USR[1] set to **0**).
- Current FIFO level can be read using TFL register.
- Programmable THRE Interrupt mode must be enabled to access “FIFO full” status using Line Status Register (LSR).

When using the “FIFO full” status, software can poll this before each write to the Transmitter FIFO. When the cts_n input becomes active (low) again, transmission resumes. Figure 8-12 illustrates a timing diagram that shows the Auto CTS operation.

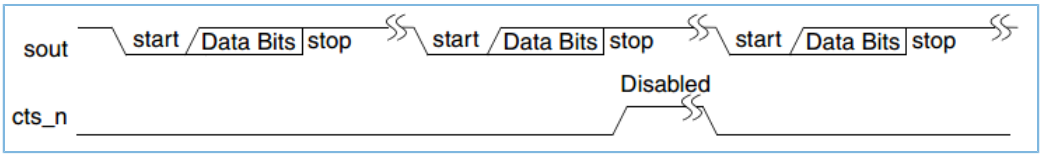

Figure 8-12 Auto CTS timing

8.4.1.4.7 Programmable THRE Interrupt

The UART can be configured to Programmable THRE Interrupt mode to increase system performance; if FIFOs are not implemented, this mode cannot be selected.

- When Programmable THRE Interrupt mode is not selected, none of the logics is implemented, and the mode cannot be enabled, reducing the overall gate counts.
- When Programmable THRE Interrupt mode is selected, it can be enabled using the Interrupt Enable Register (IER[7]).
- When FIFOs and THRE mode are implemented and enabled, the THRE Interrupts and dma_tx_req_n are active at and below a programmed transmitter FIFO empty threshold level, as opposed to empty, as shown in the flowchart in Figure 8-13.

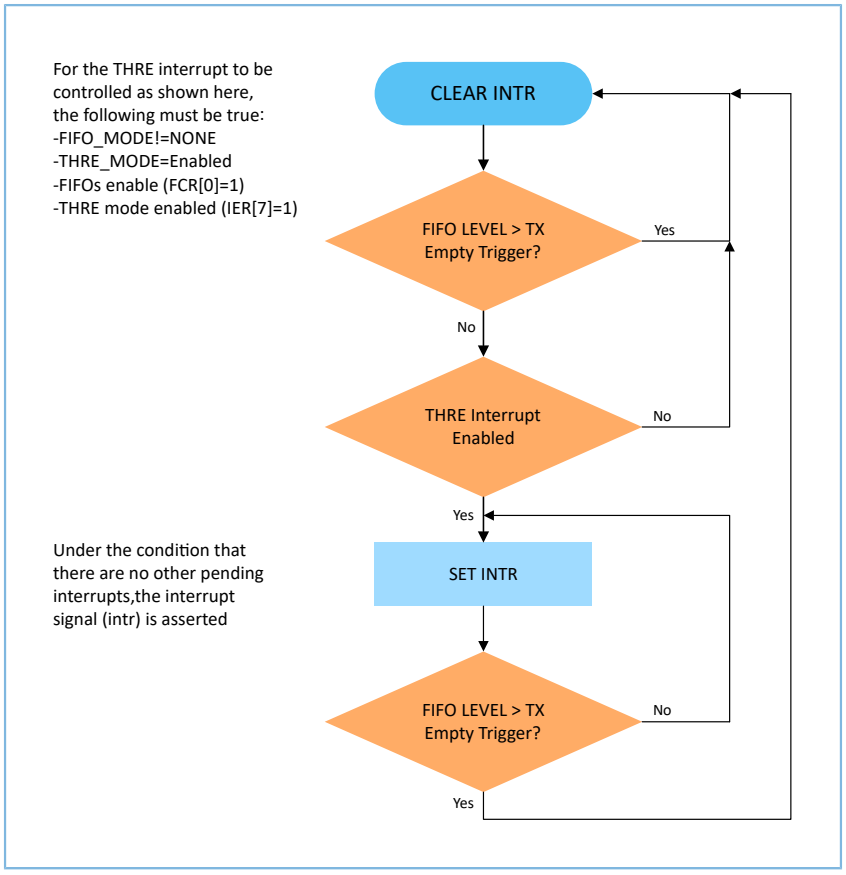

Figure 8-13 Flowchart of Interrupt Generation for Programmable THRE Interrupt Mode

The threshold level is programmed into FCR[5:4]. Available empty thresholds are:

- Empty
- 2
- 1/4
- 1/2

Selection of the best threshold value depends on the ability of the system to begin a new transmission sequence in a timely manner. However, one of these thresholds should be optimal for increasing system performance by preventing the transmitter FIFO from running empty.

In addition to the interrupt change, the Line Status Register (LSR[5]) also switches from indicating the transmitter FIFO being empty to being full. This allows software to fill the FIFO for each transmit sequence by polling LSR[5] before writing another character. The flow then allows the transmitter FIFO to be filled whenever an interrupt occurs and there is data to transmit, rather than waiting until the FIFO is completely empty. Waiting until the FIFO is empty causes a reduction in performance whenever the system is too busy to respond immediately. Further system efficiency is achieved when this mode is enabled in combination with Auto Flow Control.

Even if everything (the conditions are true for the THRE interrupt, as shown in the upper-left corner of Figure 8-13) is selected and enabled, if the FIFOs are disabled using the FCR[0] bit, the Programmable THRE Interrupt mode is also disabled. When the programmable THRE interrupt is not selected or disabled, THRE interrupts and the LSR[5] bit

function normally, signifying an empty THR or FIFO. Figure 8-14 illustrates the flowchart of THRE interrupt generation when not in programmable THRE interrupt mode.

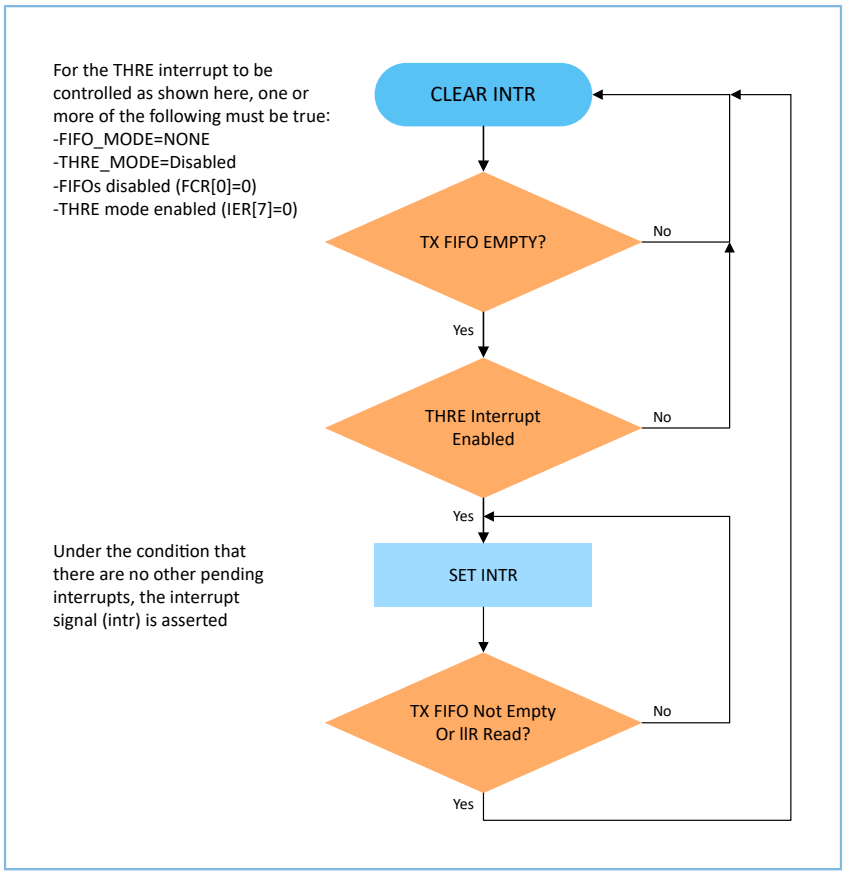

Figure 8-14 Interrupt generation when not in Programmable THRE Interrupt Mode

8.4.1.4.8 DMA Support

The UART uses two DMA channels: one for transmitting data and one for receiving data. There are two DMA modes: mode 0 and mode 1, controllable via bit 3 of the FCR register (only DMA mode 0 is available when the FIFOs are not implemented or disabled). DMA mode 0 supports single DMA data transfers at one time. DMA mode 1 supports multi-DMA data transfers, where multiple transfers are made continuously until the receiver FIFO has been emptied or the transmit FIFO has been filled. For the same DMA module, only one DMA channel is working at one time. The DMA request list of UART instance is as follows.

Table 8-69 UART-DMA handshake signal

Item	UART0	UART1	UART2	UART3	UART4	UART5
DMA0	√	√	√	√	√	
DMA1	√			√	√	√

Where, the “√” represents the UARTx can send DMAx request signal.

8.4.1.4.9 IrDA 1.0 SIR Protocol

The Infrared Data Association (IrDA) 1.0 Serial Infrared (SIR) mode supports bi-directional data communications with remote devices using infrared radiation as the transmission medium. IrDA 1.0 SIR mode specifies a maximum baud rate of 115.2 Kbaud.

The data format is similar to that of standard serial data. Each data character is sent serially in the following order:

- Begins with a start bit
- Followed by 8 data bits
- Ends with at least one stop bit

Thus, the number of data bits that can be sent is fixed. No parity information can be supplied, and only one stop bit is used in this mode. Trying to adjust the number of data bits sent or enable parity with the Line Control Register (LCR) has no effect.

You can configure the UART for IrDA 1.0 SIR to perform the following:

- Bit 6 of the Mode Control Register (MCR) enables or disables the IrDA 1.0 SIR mode.
- Disabling IrDA SIR mode causes the logic not to be implemented; the mode cannot be activated, which reduces total gate counts.
- When IrDA SIR mode is enabled and active, serial data is transmitted and received respectively on the sir_out_n and sir_in ports.

Transmission or non-transmission of a single infrared pulse indicates the following:

- Transmitting a single infrared pulse indicates logic 0.
- Non-transmission of a pulse indicates logic 1.

The width of each pulse is three sixteenths of a normal serial bit time. Thus, each new character begins with an infrared pulse for the start bit. However, received data is inverted from transmitted data due to infrared pulses energizing the photo transistor base of the IrDA receiver, which pulls its output low. This inverted transistor output is then fed to the UART sir_in port, which gives it the correct UART polarity.

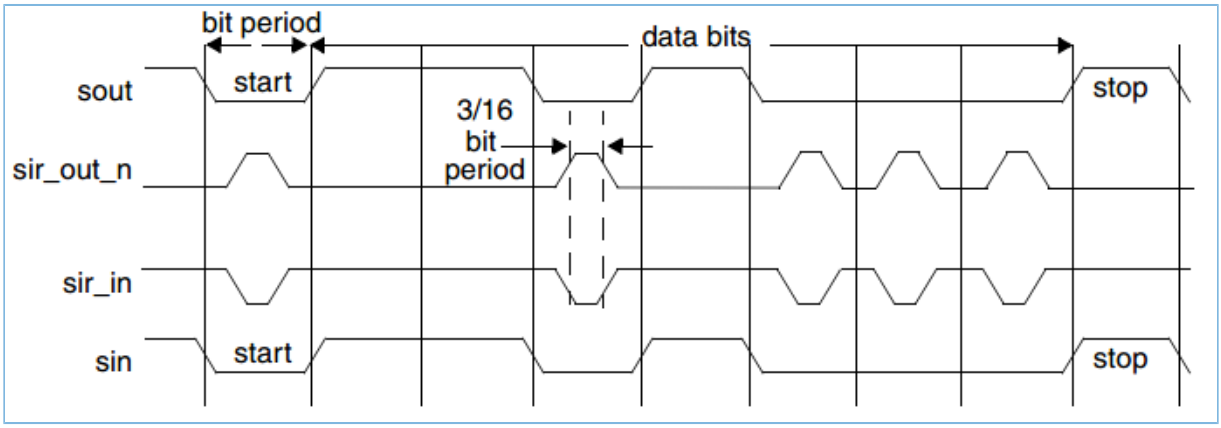

Figure 8-15 IrDA SIR data format

When IrDA SIR mode is enabled, the UART operates in a manner similar to when the mode is disabled, with one exception: data transfers can only occur in half-duplex fashion when IrDA SIR mode is enabled. This is because the IrDA

SIR physical layer specifies a minimum of 10 ms delay between transmission and reception; this 10 ms delay must be generated by software.

8.4.1.5 Registers

8.4.1.5.1 RBR

- **Name**: Receive Buffer Register
- **Description**: This register can be accessed only when the DIV_LATCH_AB bit (LCR[7]) is cleared.
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x0
- **Reset Value**: 0x00000000

Note:

*x is used to identify the UART Module instance. The “x” can be 0, 1, 2, 3, 4, or 5.

Table 8-70 Receive Buffer Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	RX_BUF	R	0x0	Receive Buffer Register. This register contains the data byte received on the serial input port (sin) in UART mode. The data in this register is valid only if the Data Ready (DATA_READY) bit in the Line Status Register (LSR)) is set.

8.4.1.5.2 DLL

- **Name**: Divisor Latch Low Register
- **Description**: This register can be accessed only when the DIV_LATCH_AB bit (LCR[7]) is set.
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x00
- **Reset Value**: 0x00000000

Table 8-71 Divisor Latch Low Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	DIV_LATCH_LOW	RW	0x0	Divisor Latch (Low).

Bits	Field Name	RW	Reset	Description
				This register makes up the lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. Note that with the Divisor Latch Registers (DIV_LATCH_LOW and DIV_LATCH_HIGH) set to zero, the baud clock is disabled and no serial communications will occur. In addition, once the DIV_LATCH_LOW is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

8.4.1.5.3 THR

- **Name**: Transmit Holding Register
- **Description**: This register can be accessed only when the DIV_LATCH_AB bit (LCR[7]) is cleared.
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x00
- **Reset Value**: 0x00000000

Table 8-72 Transmit Holding Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R	0x0	Reserved bits
7:0	TX_HDG	W	0x0	Transmit Holding Register. This register contains data to be transmitted on the serial output port (sout) in UART mode. Data should only be written to the TX_HDG when the TX_HDG Empty (TX_HDG_EMPTY) bit (LSR [5]) is set.

8.4.1.5.4 DLH

- **Name**: Divisor Latch High Register
- **Description**: This register can be accessed only when the DIV_LATCH_AB bit (LCR [7]) is set.
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x04
- **Reset Value**: 0x00000000

Table 8-73 Divisor Latch High Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	DIV_LATCH_HIGH	RW	0x0	Divisor Latch (HIGH). Upper 8-bit of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. Note that with the Divisor Latch Registers (DIV_LATCH_LOW and DIV_LATCH_HIGH) set to zero, the baud clock is disabled and no serial communications will occur. Also, once the DIV_LATCH_HIGH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

8.4.1.5.5 IER

- **Name**: Interrupt Enable Register
- **Description**: This register can be accessed only when the DIV_LATCH_AB bit (LCR [7]) is cleared.
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x04
- **Reset Value**: 0x00000000

Table 8-74 Interrupt Enable Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7	INT_GTHEE	RW	0x0	This is used to enable/disable the generation of TX_HDG_EMPTY Interrupt. **Value**: • 0x0 (DISABLED): Disable Programmable TX_HDG_EMPTY Interrupt Mode • 0x1 (ENABLED): Enable Programmable TX_HDG_EMPTY Interrupt Mode
6:4	RSVD	R		Reserved bits
3	INT_MSE	RW	0x0	Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. **Value**:

Bits	Field Name	RW	Reset	Description
				• 0x0 (DISABLED): Disable Modem Status Interrupt • 0x1 (ENABLED): Enable Modem Status Interrupt
2	INT_RX_LSE	RW	0x0	Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. **Value**: • 0x0 (DISABLED): Disable Receiver Line Status Interrupt • 0x1 (ENABLED): Enable Receiver Line Status Interrupt
1	INT_TX_HEE	RW	0x0	Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. **Value**: • 0x0 (DISABLED): Disable Transmit empty interrupt • 0x1 (ENABLED): Enable Transmit empty interrupt
0	INT_RX_EN	RW	0x0	Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFO's enabled). These are the second highest priority interrupts. **Value**: • 0x0 (DISABLED): Disable Receive data Interrupt • 0x1 (ENABLED): Enable Receive data Interrupt

8.4.1.5.6 FCR

- **Name**: FIFO Control Register
- **Description**: FIFO Control Register
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x08
- **Reset Value**: 0x00000000

Table 8-75 FIFO Control Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
7:6	RX_FIFO_TRG	W	0x0	RX Trigger (or RX_FIFO_TRG). This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt will be generated. In auto flow control mode, it is used to determine when the rts_n signal will be de-asserted only when RTC_FCT is disabled. It also determines when the dma_rx_req_n signal will be asserted in certain modes of operation. **Value**: • 0x0 (FIFO_CHAR_1): 1 character in FIFO • 0x1 (FIFO_QUARTER_FULL): FIFO 1/4 full • 0x2 (FIFO_HALF_FULL): FIFO 1/2 full • 0x3 (FIFO_FULL_2): FIFO 2 less than full
5:4	TX_EMPTY_TRG	W	0x0	TX Empty Trigger (or TX_EMPTY_TRG). This is used to select the empty threshold level at which the TX_HDG_EMPTY Interrupts will be generated when the mode is active. It also determines when the dma_tx_req_n signal will be asserted when in certain modes of operation. **Value**: • 0x0 (FIFO_EMPTY): FIFO Empty • 0x1 (FIFO_CHAR_2): 2 characters in FIFO • 0x2 (FIFO_QUARTER_FULL): FIFO 1/4 full • 0x3 (FIFO_HALF_FULL): FIFO 1/2 full
3	DMA_MODE	W	0x0	DMA Mode (or DMA_MODE). This determines the DMA signaling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected. **Value**: • 0x0 (MODE0): Mode 0 • 0x1 (MODE1): Mode 1
2	TX_FIFO_RST	W	0x0	TX FIFO Reset (or TX_FIFO_RST). This resets the control portion of the transmit FIFO and treats the FIFO as empty. This will also de-assert the DMA TX request and single signals when additional DMA handshaking signals are selected.

Bits	Field Name	RW	Reset	Description
				Value: 0x1 (RESET): Transmit FIFO reset
1	RX_FIFO_RST	W	0x0	RX FIFO Reset (or RX_FIFO_RST). This resets the control portion of the receive FIFO and treats the FIFO as empty. This will also de-assert the DMA RX request and single signals when additional DMA handshaking signals are selected. **Value**: 0x1 (RESET): Receive FIFO reset
0	FIFO_EN	W	0x0	FIFO Enable (or FIFO_EN). This enables/disables the transmit (TX) and receive (RX) FIFOs. Whenever the value of this bit is changed both the TX and RX controller portion of FIFOs is reset. **Value**: • 0x0 (DISABLED): FIFO disabled • 0x1 (ENABLED): FIFO enabled

8.4.1.5.7 IIR

- **Name**: Interrupt Identification Register
- **Description**: Interrupt Identification Register
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x08
- **Reset Value**: 0x00000001

Table 8-76 Interrupt Identification Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:6	FIFO_EN	R	0x0	FIFOs Enabled (or FIFO_EN). This is used to indicate whether the FIFOs are enabled or disabled. **Value**: • 0x0 (DISABLED): FIFOs are disabled • 0x3 (ENABLED): FIFOs are enabled
5:4	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
3:0	INT_PRIOR	R	0x1	Interrupt ID (or priority index). This indicates the highest priority pending interrupt which can be one of the following types specified in Values. **Value**: • 0x0 (MODEM_STATUS): modem status • 0x1 (NO_INTERRUPT_PENDING): no interrupt pending • 0x2 (TX_HDG _EMPTY): TX_HDG empty • 0x4 (RECEIVED_DATA_AVAILABLE): received data available • 0x6 (RECEIVER_LINE_STATUS): receiver line status • 0xC (CHARACTER_TIMEOUT): character timeout

8.4.1.5.8 LCR

- **Name**: Line Control Register
- **Description**: Line Control Register
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x0C
- **Reset Value**: 0x00000000

Table 8-77 Line Control Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7	DIV_LATCH_AB	RW	0x0	Divisor Latch Access Bit. This bit is used to enable reading and writing of the Divisor Latch Register (DIV_LATCH_LOW and DIV_LATCH_HIGH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers. **Value**: • 0x0 (DISABLED): Divisor Latch register is writable only when UART is not busy. • 0x1 (ENABLED): Divisor Latch register is always readable and writable
6	BREAK_CTRL_BIT	RW	0x0	Break Control Bit. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode,

Bits	Field Name	RW	Reset	Description
				as determined by MODEM_CTRL[4], the sout line is forced low until the Break bit is cleared. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low. **Value**: • 0x0 (DISABLED): Serial output is released for data transmission • 0x1 (ENABLED): Serial output is forced to spacing state
5	STICK_PARITY	RW	0x0	Stick Parity. This bit is used to force parity value. When PARITY_EN, EVEN_PARITY_SEL and STICK_PARITY are set to 1, the parity bit is transmitted and checked as logic 0. If PARITY_EN and Stick Parity are set to 1 and EVEN_PARITY_SEL is a logic 0, then parity bit is transmitted and checked as a logic 1. If this bit is set to 0, Stick Parity is disabled. **Value**: • 0x0 (DISABLED): STICK_PARITY disabled • 0x1 (ENABLED): STICK_PARITY enabled
4	EVEN_PARITY_SEL	RW	0x0	Even Parity Select. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If this bit is set to one, an even number of logic '1's is transmitted or checked. If set to zero, an odd number of logic '1's is transmitted or checked. **Value**: • 0x0 (ODD_PARITY): an odd parity is transmitted or checked • 0x1 (EVEN_PARITY): an even parity is transmitted or checked
3	PARITY_EN	RW	0x0	Parity Enable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. **Value**: • 0x0 (DISABLED): disable parity • 0x1 (ENABLED): enable parity

Bits	Field Name	RW	Reset	Description
2	STOP_BITS	RW	0x0	Number of Stop bits. This is used to select the number of Stop bits per character that the peripheral will transmit and receive. If set to zero, one stop bit is transmitted in the serial data. If this bit is set to one and the Data bits are set to 5 (LCR[1:0] set to zero), one and a half Stop bits are transmitted. Otherwise, two Stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver will only check the first Stop bit. **Note**: The Stop bit duration implemented by UART may appear longer due to idle time inserted between characters for some configurations and baud clock divisor values in the transmit direction. **Value**: • 0x0 (STOP_1BIT): 1 stop bit • 0x1 (STOP_1_5BIT_OR_2BIT): 1.5 stop bits when DATA_LEN_SEL (LINE_CTRL[1:0]) is zero, else 2 stop bit
1:0	DATA_LEN_SEL	RW	0x0	Data Length Select (or CLS as used in legacy). When DATA_LEN_SEL_E in LCR_EXT is set to 0, this register is used to select the number of data bits per character that the peripheral will transmit and receive. **Value**: • 0x0 (CHAR_5BITS): 5 data bits per character • 0x1 (CHAR_6BITS): 6 data bits per character • 0x2 (CHAR_7BITS): 7 data bits per character • 0x3 (CHAR_8BITS): 8 data bits per character

8.4.1.5.9 MCR

- **Name**: Modem Control Register
- **Description**: Modem Control Register
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x10
- **Reset Value**: 0x00000000

Table 8-78 Modem Control Register

Bits	Field Name	RW	Reset	Description
31:7	RSVD	R		Reserved bits
6	SIRE	RW	0x0	SIR Mode Enable Writeable only when SIR_MODE == Enabled, always readable. This is used to enable/disable the IrDA SIR Mode features. **Value**: • 0x0 (DISABLED): IrDA SIR Mode disabled. • 0x1 (ENABLED): IrDA SIR Mode enabled.
5	AFCE	RW	0x0	Auto Flow Control Enable Writeable only when AFCE_MODE == Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled. **Value**: • 0x0 (DISABLED): Auto Flow Control Mode disabled. • 0x1 (ENABLED): Auto Flow Control Mode enabled.
4	LOOP_BACK	RW	0x0	Loopback Bit. This is used to put the UART into a diagnostic mode for test purposes. **Value**: • 0x0 (DISABLED): Loopback mode disabled • 0x1 (ENABLED): Loopback mode enabled
3:2	RSVD	R		Reserved bits
1	REQ_TO_SEND	RW	0x0	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. **Value**: • 0x0 (INACTIVE): Request to Send rts_n de-asserted (logic 1) • 0x1 (ACTIVE): Request to Send rts_n asserted (logic 0)
0	RSVD	R		Reserved bits

8.4.1.5.10 LSR

- **Name**: Line Status Register
- **Description**: Line Status Register
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x14
- **Reset Value**: 0x00000060

Table 8-79 Line Status Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7	RX_FIFO_ERR	R	0x0	Receiver FIFO Error bit. This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO. **Value**: • 0x0 (NO_RX_FIFO_ERROR): No error in RX FIFO • 0x1 (RX_FIFO_ERROR): Error in RX FIFO
6	TX_EMPTY	R	0x1	Transmitter Empty bit. If FIFO is enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. **Value**: • 0x0 (DISABLED): Transmitter not empty • 0x1 (ENABLED): Transmitter empty
5	TX_HDG_EMPTY	R	0x1	Transmit Holding Register Empty bit. If TX_HDG_EMPTY mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/ enabled or not, this bit indicates that the TX_HDG or TX FIFO is empty. **Value**: • 0x0 (DISABLED): TX_HDG_EMPTY interrupt control is disabled • 0x1 (ENABLED): TX_HDG_EMPTY interrupt control is enabled
4	BREAK_INT	R	0x0	Break Interrupt bit.

Bits	Field Name	RW	Reset	Description
				This is used to indicate the detection of a break sequence on the serial input data. **Value**: • 0x0 (NO_BREAK): No break sequence detected • 0x1 (BREAK): Break sequence detected
3	FRAMING_ERR	R	0x0	Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP_BITS bit in the received data. **Value**: • 0x0 (NO_FRAMING_ERROR): no framing error • 0x1 (FRAMING_ERROR): framing error
2	PARITY_ERR	R	0x0	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PARITY_EN) bit (LCR[3]) is set. **Value**: • 0x0 (NO_PARITY_ERROR): no parity error • 0x1 (PARITY_ERROR): parity error
1	OVER_ERR	R	0x0	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. **Value**: • 0x0 (NO_OVER_RUN_ERROR): no overrun error • 0x1 (OVER_RUN_ERROR): overrun error
0	DATA_READY	R	0x0	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RX_BUF or the receiver FIFO. This bit is cleared when the receiver FIFO is empty, in the FIFO mode. **Value**: • 0x0 (NOT_READY): data not ready • 0x1 (READY): data ready

8.4.1.5.11 MSR

- **Name**: Modem Status Register
- **Description**: Whenever bit 0 is set to logic 1 to indicate a change on the modem control inputs, a modem status interrupt will be generated if enabled via the INT_EN regardless of when the change occurred. The bits (bit 0) can be set after a reset, even though their respective modem signals are inactive, because the synchronized version of the modem signals have a reset value of 0 and change to value 1 after reset. To prevent unwanted interrupts due to this change, a read of the MODEM_STAT register can be performed after reset.
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x18
- **Reset Value**: 0x00000000

Table 8-80 Modem Status Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4	CLR_SEND	R	0x0	Clear to Send. This is used to indicate the current state of the modem control line cts_n. That is, this bit is the complement cts_n. When the Clear to Send input (cts_n) is asserted, it is an indication that the modem or data set is ready to exchange data with the UART. In Loopback Mode (MODEM_CTRL[4] set to one), CLR_SEND is the same as MODEM_CTRL[1] (REQ_TO_SEND). **Value**: • 0x0 (DEASSERTED): cts_n input is de-asserted (logic 1) • 0x1 (ASSERTED): cts_n input is asserted (logic 0)
3:1	RSVD	R		Reserved bits
0	DELTA_CLR_SEND	R	0x0	Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MODEM_STAT was read. Reading the MODEM_STAT clears the DELTA_CLR_SEND bit. In Loopback Mode (MODEM_CTRL[4] set to one), DELTA_CLR_SEND reflects changes on MODEM_CTRL[1] (REQ_TO_SEND). **Note**: If the DELTA_CLR_SEND bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise),

Bits	Field Name	RW	Reset	Description
				then the DELTA_CLR_SEND bit will get set when the reset is removed if the cts_n signal remains asserted. **Value**: • 0x0 (NO_CHANGE): no change on cts_n since last read of MODEM_STAT • 0x1 (CHANGE): change on cts_n since last read of MODEM_STAT

8.4.1.5.12 SRBR

- **Name**: Shadow Receive Buffer Register
- **Description**: This register can be accessed only when the DIV_LATCH_AB bit (LINE_CTRL[7]) is cleared.
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x30
- **Reset Value**: 0x00000000

Table 8-81 Shadow Receive Buffer Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	SHADOW_RX_BUFn	R		Shadow Receive Buffer Register n. This is a shadow register for the RX_BUF and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode. The data in this register is valid only if the Data Ready (DATA_READY) bit in the Line Status Register (LINE_STAT) is set. • If FIFOs are disabled (FCR[0] set to zero), the data in the RX_BUF must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error. • If FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO will be preserved but any incoming data will be lost. An overrun error will also occur.

8.4.1.5.13 STHR

- **Name**: Shadow Transmit Holding Register
- **Description**: This register can be accessed only when the DIV_LATCH_AB bit (LINE_CTRL[7]) is cleared.
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x30
- **Reset Value**: 0x00000000

Table 8-82 Shadow Transmit Holding Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD_STHRn	R		SHADOW_TX_HDG0 31 to SHADOW_TX_HDGN_REG_SIZE Reserved bits read as 0.
7:0	SHADOW_TX_HDGn	W	0x0	Shadow Transmit Holding Register n. This is a shadow register for the TX_HDG and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains data to be transmitted on the serial output port (sout) in UART mode. Data should only be written to the TX_HDG when the TX_HDG Empty (TX_HDG_EMPTY) bit (LINE_STAT[5]) is set. • If FIFOs are disabled (FCR[0] set to zero) and TX_HDG_EMPTY is set, writing a single character to the TX_HDG clears the TX_HDG_EMPTY. Any additional writes to the TX_HDG before the TX_HDG_EMPTY is set again causes the TX_HDG data to be overwritten. • If FIFOs are enabled (FCR[0] set to one) and TX_HDG_EMPTY is set, x number of characters of data may be written to the TX_HDG before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.

8.4.1.5.14 FAR

- **Name**: FIFO Access Register
- **Description**: FIFO Access Register
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x70
- **Reset Value**: 0x00000000

Table 8-83 FIFO Access Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	FIFO_ACCESS	R	0x0	This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFO's are implemented and enabled. When FIFOs are not implemented or not enabled, it allows the RX_BUF to be written by the master and the TX_HDG to be read by the master. **Note**: When the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFO's are treated as empty. **Value**: • 0x0 (DISABLED): FIFO access mode disabled • 0x1 (ENABLED): FIFO access mode enabled

8.4.1.5.15 USR

- **Name**: UART Status Register
- **Description**: UART Status register
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x7C
- **Reset Value**: 0x00000006

Table 8-84 UART Status Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4	RX_FIFO_FULL	R	0x0	Receive FIFO Full. This is used to indicate that the receive FIFO is completely full. This bit is cleared when the RX FIFO is no longer full. **Value**: • 0x0 (NOT_FULL): Receive FIFO not full • 0x1 (FULL): Receive FIFO full
3	RX_FIFO_EMPTY	R	0x0	Receive FIFO Not Empty.

Bits	Field Name	RW	Reset	Description
				This is used to indicate that the receive FIFO contains one or more entries. This bit is cleared when the RX FIFO is empty. **Value**: • 0x0 (EMPTY): Receive FIFO is empty • 0x1 (NOT_EMPTY): Receive FIFO is not empty
2	TX_FIFO_EMPTY	R	0x1	Transmit FIFO Empty. This is used to indicate that the transmit FIFO is completely empty. This bit is cleared when the TX FIFO is no longer empty. **Value**: • 0x0 (NOT_EMPTY): Transmit FIFO is not empty • 0x1 (EMPTY): Transmit FIFO is empty
1	TX_FIFO_FULL	R	0x1	Transmit FIFO Not Full. This is used to indicate that the transmit FIFO in not full. This bit is cleared when the TX FIFO is full. **Value**: • 0x0 (FULL): Transmit FIFO is full • 0x1 (NOT_FULL): Transmit FIFO is not full
0	RSVD	R		Reserved bit

8.4.1.5.16 TFL

- **Name**: Transmit FIFO Level
- **Description**: Transmit FIFO Level
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x80
- **Reset Value**: 0x00000000

Table 8-85 Transmit FIFO Level

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	TX_FIFO_LEVEL	R	0x0	Transmit FIFO Level. This indicates the number of data entries in the transmit FIFO.

8.4.1.5.17 RFL

- **Name**: Receive FIFO Level
- **Description**: Receive FIFO Level
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x84
- **Reset Value**: 0x00000000

Table 8-86 Receive FIFO Level

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	RX_FIFO_LEVEL	R	0x0	Receive FIFO Level. This is indicates the number of data entries in the receive FIFO.

8.4.1.5.18 SRR

- **Name**: Software Reset Register
- **Description**: Software Reset Register
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x88
- **Reset Value**: 0x00000000

Table 8-87 Software Reset Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	TX_FIFO_RST	W	0x0	TX FIFO Reset. This is a shadow register for the TX FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FIFO_CTRL values (which are pretty static) just to reset the transmit FIFO
1	RX_FIFO_RST	W	0x0	RX FIFO Reset. This is a shadow register for the RX FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FIFO_CTRL values (which are pretty static) just to reset the receive FIFO.
0	UART_RST	W	0x0	UART Reset.

Bits	Field Name	RW	Reset	Description
				This asynchronously resets the UART and synchronously removes the reset assertion. For a two-clock implementation both PCLK and SCLK domains will be reset. **Value**: • 0x0 (NO_RESET): No UART Reset • 0x1 (RESET): UART reset

8.4.1.5.19 SRTS

- **Name**: Shadow Request to Send Register
- **Description**: Shadow Request to Send
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x8C
- **Reset Value**: 0x00000000

Table 8-88 Shadow Request to Send

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	SHADOW_REQ_SEND	RW	0x0	Shadow Request to Send. This is a shadow register for the REQ_TO_SEND bit (MODEM_CTRL [1]), this can be used to remove the burden of having to performing a read modify write on the MODEM_CTRL. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. **Value**: • 0x0 (DEASSERTED): Shadow Request to Send uart_rts_n logic 1 • 0x1 (ASSERTED): Shadow Request to Send uart_rts_n logic 0

8.4.1.5.20 SBCR

- **Name**: Shadow Break Control Register
- **Description**: Shadow Break Control Register

- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x90
- **Reset Value**: 0x00000000

Table 8-89 Shadow Break Control Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	SHADOW_BREAK_CB	RW	0x0	Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]). This can be used to remove the burden of having to performing a read modify write on the LINE_CTRL. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MODEM_CTRL [4], the sout line is forced to be low until the Break bit is cleared. **Value**: • 0x0 (NO_BREAK): No spacing on serial output • 0x1 (BREAK): Serial output forced to the spacing

8.4.1.5.21 SFE

- **Name**: Shadow FIFO Enable Register
- **Description**: Shadow FIFO Enable Register
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x98
- **Reset Value**: 0x00000000

Table 8-90 Shadow FIFO Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	SHADOW_FIFO_EN	RW	0x0	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FIFO_CTRL in memory and having to mask this value so that only the FIFO enable bit gets updated. This enables/disables the transmit (TX) and

Bits	Field Name	RW	Reset	Description
				receive (RX) FIFO's. If this bit is set to zero (disabled) after being enabled then both the TX and RX controller portions of FIFO's will be reset. **Value**: • 0x0 (DISABLED): FIFOs are disabled • 0x1 (ENABLED): FIFOs are enabled

8.4.1.5.22 SRT

- **Name**: Shadow RX Trigger Register
- **Description**: Shadow RX Trigger Register
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0x9C
- **Reset Value**: 0x00000000

Table 8-91 Shadow RCVR Trigger Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1:0	SHADOW_RX_TRG	RW	0x0	Shadow RX Trigger. This is a shadow register for the RX trigger bits (FCR[7:6]). This can be used to remove the burden of having to store the previously written value to the FIFO_CTRL in memory and having to mask this value so that only the RX trigger bit gets updated. **Value**: • 0x0 (FIFO_CHAR_1): 1 character in FIFO • 0x1 (FIFO_QUARTER_FULL): FIFO 1/4 full • 0x2 (FIFO_HALF_FULL): FIFO 1/2 full • 0x3 (FIFO_FULL_2): FIFO 2 less than full

8.4.1.5.23 STET

- **Name**: Shadow TX Empty Trigger Register
- **Description**: Shadow TX Empty Trigger Register
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0xA0

- **Reset Value**: 0x00000000

Table 8-92 Shadow TX Empty Trigger Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1:0	SHADOW_TX_TRG	RW	0x0	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]). This can be used to remove the burden of having to store the previously written value to the FIFO_CTRL in memory and having to mask this value so that only the TX empty trigger bit gets updated. **Value**: • 0x0 (FIFO_EMPTY): FIFO empty • 0x1 (FIFO_CHAR_2): 2 characters in FIFO • 0x2 (FIFO_QUARTER_FULL): FIFO 1/4 full • 0x3 (FIFO_HALF_FULL): FIFO 1/2 full

8.4.1.5.24 DLF

- **Name**: Divisor Latch Fraction Register
- **Description**: Divisor Latch Fraction Register
- **Base Address**: 0x4000C500 + x*0x100
- **Offset**: 0xC0
- **Reset Value**: 0x00000000

Table 8-93 Divisor Latch Fraction Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	DIV_LATCH_FRACTION	RW	0x0	Fractional part of divisor. The fractional value is added to integer value set by DIV_LATCH_HIGH and DIV_LATCH_LOW. Fractional value is determined by (Divisor Fraction value)/(2^DLF_SIZE).

8.4.1.6 Electrical Specifications

Table 8-94 Baud rate with 96 MHz Serial Clock

Baud Rate	Divider	DLH Reg	DLL Reg	DLF Reg	Actual Baud Rate	Error (%)
1200	5000	19	136	0	1200	0
2400	2500	9	196	0	2400	0
4800	1250	4	226	0	4800	0
9600	626	2	113	0	9600	0
14400	416.67	1	160	11	14399.28	0
19200	312.50	1	56	8	19200	0
28800	208.33	0	208	5	28802.88	0.01
38400	156.25	0	156	4	38400	0
57600	104.17	0	104	3	57588.48	0.02
115200	52.08	0	52	1	115246.10	0.04
230400	26.04	0	26	4	230215.83	0.08
460800	13.021	0	13	0	461538.46	0.16
921600	6.51	0	6	8	923076.92	0.16
1000000	6	0	6	0	1000000	0
2000000	3	0	3	0	2000000	0
3000000	2	0	2	0	3000000	0
4000000	1.5	0	1	8	4000000	0

8.4.2 SPI

8.4.2.1 Introduction

GR5526 SoC has two Serial Peripheral Interface (SPI) instances.

- 1 SPI Master (SPIM) with two slave select lines that can interface to up to two slaves that share the same lines with separate selects.
- 1 SPI Slave (SPIS) that can connect to a single master.

8.4.2.2 Main Features

Main features of the SPIM/SPIS modules include:

- Motorola SPI support
- 2 slave select lines for master mode to support up to 2 slaves.
- Built-in 16 word RX/TX FIFOs for continuous SPI bursts.
- Transfer data size up to 32 bits.

- Programmable output interface frequency up to 48 MHz for SPIM and 16 MHz for SPIS.
- Supports SPI four modes for different clock edge and phase configurations.
- DMA support
- Maskable interrupt generation

8.4.2.3 Block Diagram

Figure 8-16 shows the SPI Controller with the following functions and interfaces:

- APB interface and DMA Controller Interface
- Transmit and receive FIFO controllers and an FSM controller
- Register block
- Shift control and interrupt logic

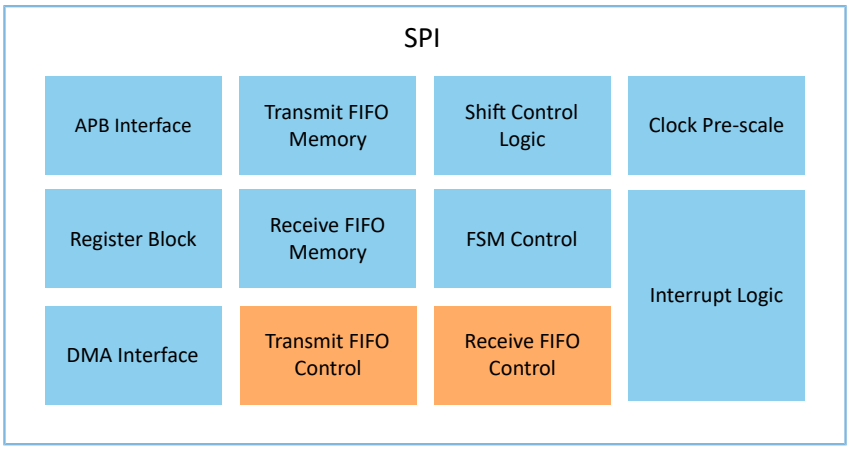

Figure 8-16 SPI block diagram

8.4.2.4 Functional Description

8.4.2.4.1 Motorola Serial Peripheral Interface

With the SPI, the clock polarity (SCPOL) configuration parameter determines whether the inactive state of the serial clock is high or low. To transmit data, both SPI peripherals must have identical serial clock phase (SCPH, control by CTRL0[6]) and clock polarity (SCPOL, control by CTRL0[7]) values. The data frame can be 4 bits to 32 bits.

When the configuration parameter SCPH = 0, data transmission begins on the falling edge of the slave select signal. The first data bit is captured by the master and slave peripherals on the first edge of the serial clock; therefore, valid data must be present on the txd and rxd lines prior to the first serial clock edge.

Figure 8-17 shows a timing diagram for a single SPI data transfer with SCPH = 0 (CTRL0[6] = 0). The serial clock is shown for configuration parameters SCPOL = 0 (CTRL0[7] = 0) and SCPOL = 1 (CTRL0[7] = 1).

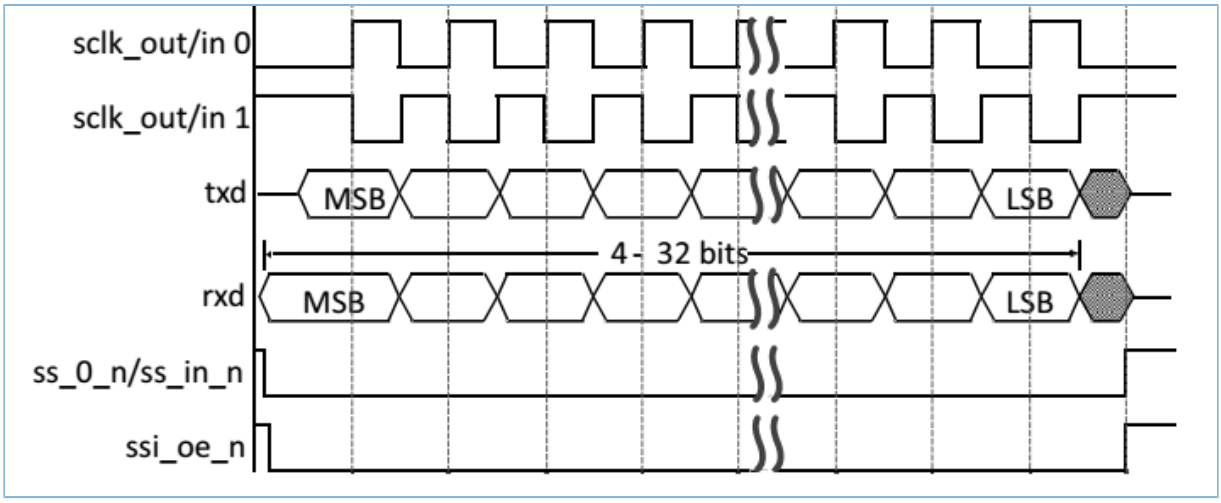

Figure 8-17 SPI serial format (SCPH = 0)

The following signals are illustrated in the timing diagrams in this section:

- sclk_out – serial clock from SPI block master (master configuration only)
- sclk_in – serial clock from SPI block slave (slave configuration only)
- txd – transmit data line for the SPI block master/slave
- rxd – receive data line for the SPI block master/slave
- ss_0_n – slave select signal from SPI block master (master configuration only)
- ss_in_n – slave select input to the SPI block slave
- ss_oe_n – output enable for the SPI block master/slave

Two different modes of continuous data transfers are supported when SCPH = 0; the selection of the desired operation mode is done by configuring bit24 (S_ST_EN) of the CTRL0 register:

- When CTRL0[24] is programmed to 1, SPI block toggles the slave select signal between frames, and the serial clock is held to its default value while the slave select signal is active; this operating mode is illustrated in Figure 8-18.

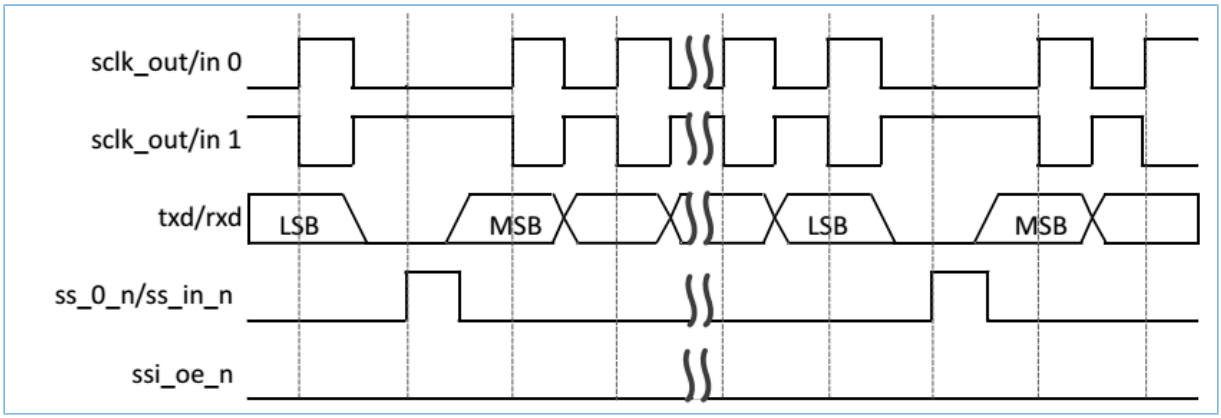

Figure 8-18 Serial format continuous transfers (SCPH = 0) when CTRL0[24] = 1

- When CTRL0[24] is programmed to 0, the slave select signal stays low and the serial clock runs continuously for the duration of the transfer; this operating mode is illustrated in Figure 8-19.

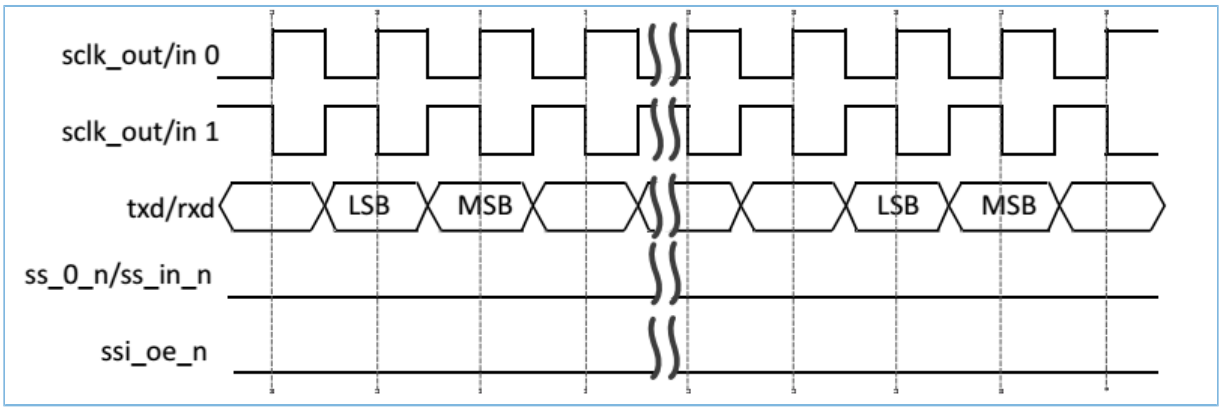

Figure 8-19 Serial format continuous transfers (SCPH = 0) when CTRL0[24] = 0

When the configuration parameter SCPH = 1 (CTRL0[6] = 1), both master and slave peripherals begin transmitting data on the first serial clock edge after the slave select line is activated. The first data bit is captured on the second (trailing) serial clock edge. Data is propagated by the master and slave peripherals on the leading edge of the serial clock. During continuous data frame transfers, the slave select line may be held active-low until the last bit of the last frame has been captured.

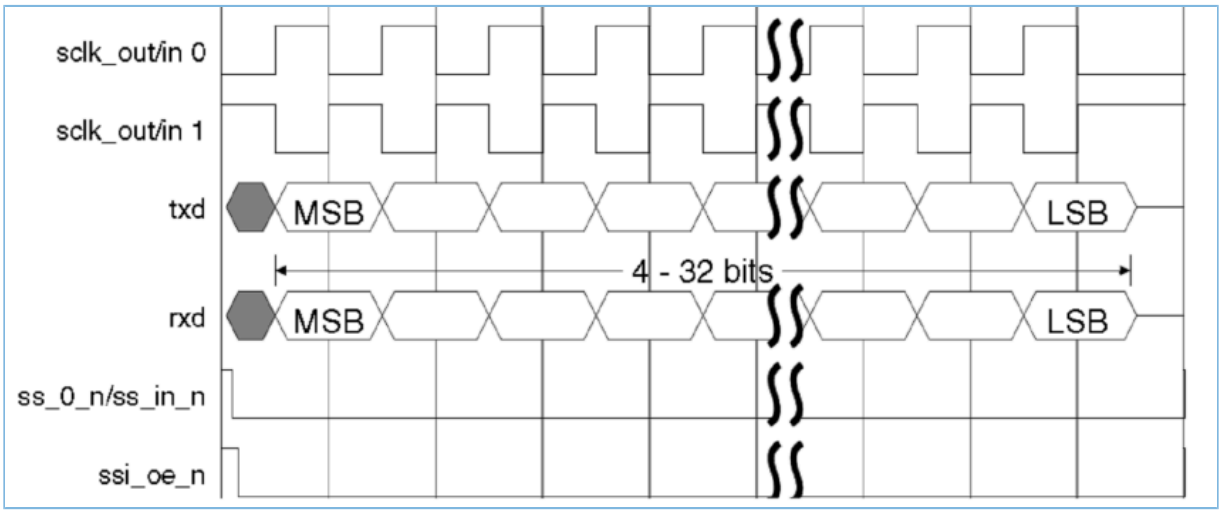

Figure 8-20 SPI serial format (SCPH = 1)

8.4.2.4.2 Transmit and Receive FIFO Buffers

The width of both transmit and receive FIFO buffers is fixed at 32 bits. Data frames that are less than 32 bits must be right-justified when written into the transmit FIFO buffer. The shift control logic automatically right-justifies received data in the receive FIFO buffer.

Each data entry in the FIFO buffers contains a single data frame. It is not possible to store multiple data frames in a single FIFO location; for example, users may not store two 8-bit data frames in a single FIFO location. If an 8-bit data frame is required, the upper 8-bits of the FIFO entry are ignored or unused when the serial shifter transmits the data.

Transmit FIFO

The transmit FIFO is loaded by writing to the SPI Controller data register (DATA). Data is popped from the transmit FIFO by the shift control logic into the transmit shift register. The transmit FIFO generates a FIFO empty interrupt request (ssi_txe_intr) when the number of entries in the FIFO is less than or equal to the FIFO threshold value. The threshold value, set through the programmable transmit FIFO threshold level register (TX_FIFO_TL[2:0]), determines the level of FIFO entries at which an interrupt is generated. The threshold value allows users to provide early indication to the processor that the transmit FIFO is nearly empty. A transmit FIFO overflow interrupt (ssi_txo_intr) is generated if users attempt to write data into an already full transmit FIFO.

Receive FIFO

Data is popped from the receive FIFO by reading the SPI Controller data register (DATA). The receive FIFO is loaded from the receive shift register by the shift control logic. The receive FIFO generates a FIFO-full interrupt request (ssi_rxf_intr) when the number of entries in the FIFO is greater than or equal to the FIFO threshold value plus 1. The threshold value, set through programmable receive FIFO threshold level register (RX_FIFO_TL[2:0]), determines the level of FIFO entries at which an interrupt is generated.

The threshold value allows users to provide early indication to the processor that the receive FIFO is nearly full. A receive FIFO overrun interrupt (ssi_rxo_intr) is generated when the receive shift logic attempts to load data into a completely full receive FIFO. However, this newly received data is lost. A receive FIFO underflow interrupt (ssi_rxu_intr) is generated if users attempt to read from an empty receive FIFO. This alerts the processor that the read data is invalid.

8.4.2.4.3 SPI Interrupts

The SPI Controller supports combined and individual interrupt requests, each of which can be masked. The combined interrupt request is the OR'ed result of all other SPI Controller interrupts after masking. The system designer has the choice of routing individual interrupt requests or only the combined interrupt request to the Interrupt Controller. All SPI Controller interrupts are level interrupts that share the same active polarity level; users can configure this polarity level as active-high or active-low.

The SPI Controller interrupts are described as follows:

- Transmit FIFO Empty Interrupt (ssi_txe_intr) – Set when the transmit FIFO is equal to or below its threshold value and requires service to prevent an under-run. The threshold value, set through a software-programmable register, determines the level of transmit FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data is written into the transmit FIFO buffer, bringing it over the threshold level.
- Transmit FIFO Overflow Interrupt (ssi_txo_intr) – Set when an APB access attempts to write into the transmit FIFO after it has been completely filled. When set, data written from the APB is discarded. This interrupt remains set until users read the transmit FIFO overflow interrupt clear register (TX_FIFO_OIC[0]).
- Receive FIFO Full Interrupt (ssi_rxf_intr) – Set when the receive FIFO is equal to or above its threshold value plus 1 and requires service to prevent an overflow. The threshold value, set through a software-programmable register, determines the level of receive FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data is read from the receive FIFO buffer, bringing it below the threshold level.
- Receive FIFO Overflow Interrupt (ssi_rxo_intr) – Set when the receive logic attempts to place data into the receive FIFO after it has been completely filled. When set, newly received data is discarded. This interrupt remains set until users read the receive FIFO overflow interrupt clear register (RX_FIFO_OIC[0]).

- Receive FIFO Underflow Interrupt (ssi_rxu_intr) – Set when an APB access attempts to read from the receive FIFO when it is empty. When set, zeros are read back from the receive FIFO. This interrupt remains set until users read the receive FIFO underflow interrupt clear register (RX_FIFO_UIC[0]).

8.4.2.4.4 Transfer Modes

When transferring data on the serial bus, the SPI Controller operates in the modes discussed in this section. The transfer mode (TMOD) is set by writing to bit8 and bit9 of the SPI control register 0 (CTRL0).

Note:

The transfer mode setting does not affect the duplex of the serial transfer.

8.4.2.4.4.1 Transmit and Receive

When CTRL0[9:8] = 0x0, both transmit and receive logics are valid. The data transfer occurs as normal according to the selected frame format (serial protocol). Transmit data is popped from the transmit FIFO and sent through the txd (tx data) line to the target device, which replies with data on the rxd (rx data) line. The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame.

8.4.2.4.4.2 Transmit Only

When CTRL0[9:8] = 0x1, the receive data is invalid and should not be stored in the receive FIFO. The data transfer occurs as normal, according to the selected frame format (serial protocol). Transmit data is popped from the transmit FIFO and sent through the txd line to the target device, which replies with data on the rxd line. At the end of the data frame, the receive shift register does not load its newly received data into the receive FIFO. The data in the receive shift register is overwritten by the next transfer. Users should mask interrupts originating from the receive logic when this mode is entered.

8.4.2.4.4.3 Receive Only

When CTRL0[9:8] = 0x2, the transmit data is invalid. When configured as a slave, the transmit FIFO is never popped in Receive Only mode. The txd output remains at a constant logic level during the transmission. The data transfer occurs as normal according to the selected frame format (serial protocol). The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame. Users should mask interrupts originating from the transmit logic when this mode is entered.

8.4.2.4.4.4 EEPROM Read

Note:

This transfer mode is only valid for master configurations.

When CTRL0[9:8] = 0x3, the transmit data is used to transmit an opcode and/or an address to the EEPROM device. Typically, this takes three data frames (8-bit opcode followed by 8-bit upper address and 8-bit lower address). During the transmission of the opcode and address, no data is captured by the receive logic (as long as the SPI Controller master is transmitting data on its txd line, data on the rxd line is ignored). The SPI Controller master continues to transmit data until the transmit FIFO is empty. Therefore, users should only have enough data frames in the transmit

FIFO to supply the opcode and address to the EEPROM. If more data frames are in the transmit FIFO than needed, then read data is lost.

When the transmit FIFO becomes empty (all control information has been sent), data on the receive line (rxd) is valid and is stored in the receive FIFO; the txd output is held at a constant logic level. The serial transfer continues until the number of data frames received by the SPI Controller master matches the value of the NDF (Number of Data Frames) field in the SPI Control register 1 (CTRL1[15:0]) plus 1.

8.4.2.4.5 SPI Master Operation

Data transfers are started by the serial-master device. When the SPI Controller is enabled (SPI Enable Register, SSI_EN[0]=1), at least one valid data entry is present in the transmit FIFO and a serial-slave device is selected. When actively transferring data, the busy flag (BUSY) in the status register (STAT[0]) is set. Users must wait until the busy flag is cleared before attempting a new serial transfer.

When the transfer mode is “transmit and receive” or “transmit only” (CTRL0[9:8] = 0x0 or CTRL0[9:8] = 0x1, respectively), transfers are terminated by the shift control logic when the transmit FIFO is empty. For continuous data transfers, users must ensure that the transmit FIFO buffer does not become empty before all the data has been transmitted. The transmit FIFO threshold level (TX_FIFO_TL[2:0]) can be used to early interrupt (ssi_txe_intr) the processor, indicating that the transmit FIFO buffer is nearly empty. When a DMA is used for APB accesses, the DMA transmit data level (DMA_TX_DL[2:0]) can be used to early request (dma_tx_req) the DMA Controller, indicating that the transmit FIFO is nearly empty. The FIFO can then be refilled with data to continue the serial transfer. Users may also write a block of data (at least two FIFO entries) into the transmit FIFO before enabling a serial slave. This ensures that serial transmission does not begin until the number of data frames that make up the continuous transfer are present in the transmit FIFO.

When the transfer mode is “receive only” (CTRL0[9:8] = 0x2), a serial transfer is started by writing one “dummy” data word into the transmit FIFO when a serial slave is selected. The TXD output from the SPI Controller is held at a constant logic level for the duration of the serial transfer. The transmit FIFO is popped only once at the beginning and may remain empty for the duration of the serial transfer. The end of the serial transfer is controlled by the “number of data frames” (NDF) field in control register 1 (CTRL1[15:0]).

If, for example, users want to receive 24 data frames from a serial-slave peripheral, you should program the NDF field with the value 23; the receive logic terminates the serial transfer when the number of frames received is equal to the NDF value + 1. This transfer mode increases the bandwidth of the APB bus as the transmit FIFO never needs to be serviced during the transfer. The receive FIFO buffer should be read each time the receive FIFO generates a FIFO full interrupt request to prevent an overflow.

When the transfer mode is “eeprom_read” (CTRL0[9:8] = 0x3), a serial transfer is started by writing the opcode and/or address into the transmit FIFO when a serial slave (EEPROM) is selected. The opcode and address are transmitted to the EEPROM device, after which read data is received from the EEPROM device and stored in the receive FIFO. The end of the serial transfer is controlled by the NDF field in the control register 1 (CTRL1[15:0]).

The receive FIFO threshold level (RX_FIFO_TL[2:0]) can be used to give early indication that the receive FIFO is nearly full. When a DMA is used for APB accesses, the DMA receive data level (DMA_RX_DL[2:0]) can be used to early request (dma_rx_req) the DMA Controller, indicating that the receive FIFO is nearly full.

8.4.2.4.6 SPI Slave Operation

If the SPI Controller slave is receive only (CTRL0[9:8] = 0x2), the transmit FIFO does not need to contain valid data because the data currently in the transmit shift register is resent each time the slave device is selected. The TX_ERR error flag in the status register (STAT[5]) is not set when CTRL0[9:8] = 0x1. Users should mask the transmit FIFO empty interrupt when this mode is used.

If the SPI Controller slave transmits data to the master, users must ensure that data exists in the transmit FIFO before a transfer is initiated by the serial-master device. If the master initiates a transfer to the SPI Controller slave when no data exists in the transmit FIFO, an error flag (TX_ERR) is set in the status register (STAT[5]), and the previously transmitted data frame is resent on txd. For continuous data transfers, users must ensure that the transmit FIFO buffer does not become empty before all the data has been transmitted. The transmit FIFO threshold level register (TX_FIFO_TL[2:0]) can be used to early interrupt (ssi_txe_intr) the processor, indicating that the transmit FIFO buffer is nearly empty. When a DMA Controller is used for APB accesses, the DMA transmit data level register (DMA_TX_DL[2:0]) can be used to early request (dma_tx_req) the DMA Controller, indicating that the transmit FIFO is nearly empty. The FIFO can then be refilled with data to continue the serial transfer. The receive FIFO buffer should be read each time the receive FIFO generates a FIFO full interrupt request to prevent an overflow. The receive FIFO threshold level register (RX_FIFO_TL[2:0]) can be used to give early indication that the receive FIFO is nearly full. When a DMA Controller is used for APB accesses, the DMA receive data level register (DMA_RX_DL[2:0]) can be used to early request (dma_rx_req) the DMA controller, indicating that the receive FIFO is nearly full.

8.4.2.5 Registers

8.4.2.5.1 CTRL0

- **Name:** SPI Control Register 0
- **Description:** This register controls the serial data transfer. It is impossible to write to this register when the SPI is enabled.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x00
- **Reset Value:** 0x01070000

Table 8-95 Control Register 0

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits.
24	S_ST_EN	RW	0x1	Slave Select Toggle Enable. When operating in SPI mode with clock phase (SCPH) set to 0, this register controls the behavior of the slave select line (CS) between data frames. If this register field is set to 1 the CS line will toggle between consecutive data frames, with the serial clock (sclk) being held to its default value while CS is high; if this register field is set to 0 the CS will stay low and sclk will run continuously for the duration of the transfer.

Bits	Field Name	RW	Reset	Description
				Note: When the SPI is configured as a slave, this register serves no purpose.
23:21	RSVD	R		Reserved bits.
20:16	DATA_FRAME_SIZE	RW	0x7	Data Frame Size in 32-bit transfer size mode. Used to select the data frame size in 32-bit transfer mode. When the data frame size is programmed to be less than 32 bits, the receive data is automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You are responsible for making sure that transmit data is right-justified before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. **Value:** • 0x3 (FRAME_04BITS): 4-bit serial data transfer • 0x4 (FRAME_05BITS): 5-bit serial data transfer • 0x5 (FRAME_06BITS): 6-bit serial data transfer • 0x6 (FRAME_07BITS): 7-bit serial data transfer • 0x7 (FRAME_08BITS): 8-bit serial data transfer • 0x8 (FRAME_09BITS): 9-bit serial data transfer • 0x9 (FRAME_10BITS): 10-bit serial data transfer • 0xa (FRAME_11BITS): 11-bit serial data transfer • 0xb (FRAME_12BITS): 12-bit serial data transfer • 0xc (FRAME_13BITS): 13-bit serial data transfer • 0xd (FRAME_14BITS): 14-bit serial data transfer • 0xe (FRAME_15BITS): 15-bit serial data transfer • 0xf (FRAME_16BITS): 16-bit serial data transfer • 0x10 (FRAME_17BITS): 17-bit serial data transfer • 0x11 (FRAME_18BITS): 18-bit serial data transfer • 0x12 (FRAME_19BITS): 19-bit serial data transfer • 0x13 (FRAME_20BITS): 20-bit serial data transfer • 0x14 (FRAME_21BITS): 21-bit serial data transfer • 0x15 (FRAME_22BITS): 22-bit serial data transfer • 0x16 (FRAME_23BITS): 23-bit serial data transfer

Bits	Field Name	RW	Reset	Description
				• 0x17 (FRAME_24BITS): 24-bit serial data transfer • 0x18 (FRAME_25BITS): 25-bit serial data transfer • 0x19 (FRAME_26BITS): 26-bit serial data transfer • 0x1a (FRAME_27BITS): 27-bit serial data transfer • 0x1b (FRAME_28BITS): 28-bit serial data transfer • 0x1c (FRAME_29BITS): 29-bit serial data transfer • 0x1d (FRAME_30BITS): 30-bit serial data transfer • 0x1e (FRAME_31BITS): 31-bit serial data transfer • 0x1f (FRAME_32BITS): 32-bit serial data transfer
15:12	CTRL_FRAME_SIZE	RW	0x0	Control Frame Size. Selects the length of the control word for the Microwire frame format. **Value:** • 0x0 (SIZE_01_BIT): 1-bit Control Word • 0x1 (SIZE_02_BIT): 2-bit Control Word • 0x2 (SIZE_03_BIT): 3-bit Control Word • 0x3 (SIZE_04_BIT): 4-bit Control Word • 0x4 (SIZE_05_BIT): 5-bit Control Word • 0x5 (SIZE_06_BIT): 6-bit Control Word • 0x6 (SIZE_07_BIT): 7-bit Control Word • 0x7 (SIZE_08_BIT): 8-bit Control Word • 0x8 (SIZE_09_BIT): 9-bit Control Word • 0x9 (SIZE_10_BIT): 10-bit Control Word • 0xa (SIZE_11_BIT): 11-bit Control Word • 0xb (SIZE_12_BIT): 12-bit Control Word • 0xc (SIZE_13_BIT): 13-bit Control Word • 0xd (SIZE_14_BIT): 14-bit Control Word • 0xe (SIZE_15_BIT): 15-bit Control Word • 0xf (SIZE_16_BIT): 16-bit Control Word
11	SHIFT_REG_LOOP	RW	0x0	Shift Register Loop. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (NORMAL_MODE): Normal mode operation • 0x1 (TESTING_MODE): Test mode: TX & RX shift register connected
10	S_OUT_EN	RW	0x0	Slave Output Enable. Relevant only when the SPI is configured as a serial-slave device. When configured as a serial master, this bit field has no functionality. **Value:** • 0x0 (ENABLED): Slave Output is enabled • 0x1 (DISABLED): Slave Output is disabled **Note:** When the SPI is configured as a master, this register serves no purpose.
9:8	XFE_MODE	RW	0x0	Transfer Mode. Selects the mode of transfer for serial communication. This field does not affect transfer duplicity, and only indicates whether the received or transmitted data is valid. • 0x0 (TX_AND_RX): Transmit & Receive mode In this mode, both transmitted data and received data are valid. Transfer continues until the transmit FIFO is empty. Data received from an external device is stored into the receive FIFO memory, which can be accessed by the host processor. • 0x1 (TX_ONLY): Transmit only mode In this mode, data received from an external device is not valid and is not stored in the receive FIFO memory; it will be overwritten in the next transfer. • 0x2 (RX_ONLY): Receive only mode In this mode, transmitted data is not valid. After the first write to the transmit FIFO, the same word is retransmitted during transfer. • 0x3 (EEPROM_READ): EEPROM readmode In this mode, received data is not valid while control data is being transmitted. When all control data is sent to the EEPROM, received data becomes valid and transmitted data becomes invalid. All data in the transmit FIFO is considered control data in this mode. This transfer mode is only valid when the SPI is configured as master device.
7	SERIAL_CLK_POL	RW	0x0	Serial Clock Polarity.

Bits	Field Name	RW	Reset	Description
				Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the SPI master is not actively transferring data on the serial bus. **Value:** • 0x0 (SCLK_LOW): Inactive state of serial clock is low • 0x1 (SCLK_HIGH): Inactive state of serial clock is high
6	SERIAL_CLK_PHASE	RW	0x0	Serial Clock Phase. Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data is captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data is captured on the second edge of the serial clock. **Value:** • 0x0 (SCPH_MIDDLE): Serial clock toggles in middle of first data bit • 0x1 (SCPH_START): Serial clock toggles at start of first data bit
5:4	FRAME_FORMAT	RW	0x0	Frame Format. Selects which serial protocol transfers the data. **Value:** • 0x0 (MOTOROLA_SPI): Motorola SPI Frame Format • 0x1 (TEXAS_SSP): Texas Instruments SSP Frame Format • 0x2 (NS_MICROWIRE): National Microwire Frame Format • 0x3 (RESERVED): Reserved value
3:0	RSVD	R		Reserved bits.

8.4.2.5.2 CTRL1

- **Name:** SPI Control Register 1
- **Description:** This register exists only when the SPI is configured as a master device. Control register 1 controls the end of serial transfers in receive-only mode. It is impossible to write to this register when the SPI is enabled.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x04

- **Reset Value:** 0x00000000

Table 8-96 Control Register 1

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits.
15:0	NUM_DATA_FRAME	RW	0x0	Number of Data Frames. When CTRL0[9:8] = 0x2 or CTRL0[9:8] = 0x3, this register field sets the number of data frames to be continuously received by the SPI. The SPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64KB of data in a continuous transfer. **Note:** When the SPI is configured as a slave, this register serves no purpose.

8.4.2.5.3 SSI_EN

- **Name:** SPI Enable Register
- **Description:** This register enables and disables the SPI.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x08
- **Reset Value:** 0x00000000

Table 8-97 SSI Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits.
0	SSI_EN	RW	0x0	SSI Enable. Enables and disables all SPI operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. **Value:** • 0x0 (DISABLE): Disables Serial Transfer • 0x1 (ENABLED): Enables Serial Transfer

8.4.2.5.4 MW_CTRL

- **Name:** Microwire Control Register

- **Description:** This register controls the direction of the data word for the half-duplex Microwire serial protocol. It is impossible to write to this register when the SPI is enabled.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x0C
- **Reset Value:** 0x00000000

Table 8-98 Microwire Control Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits.
2	MW_HSG	RW	0x0	Microwire Handshaking. Used to enable and disable the busy/ready handshaking interface for the Microwire protocol. When enabled, the SPI checks for a ready status from the target slave, after the transfer of the last data/control bit, before clearing the BUSY status in the STAT register. **Value:** • 0x0 (DISABLE): Handshaking interface is disabled • 0x1 (ENABLED): Handshaking interface is enabled **Note:** When the SPI is configured as a slave, this register serves no purpose.
1	MW_DIR_DW	RW	0x0	Microwire Control. Defines the direction of the data word when the Microwire serial protocol is used. **Value:** • 0x0 (RECEIVE): SPI receives data • 0x1 (TRANSMIT): SPI transmits data
0	MW_XFE_MODE	RW	0x0	Microwire Transfer Mode. Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. **Value:** • 0x0 (NON_SEQUENTIAL): Non-Sequential Microwire Transfer

Bits	Field Name	RW	Reset	Description
				• 0x1 (SEQUENTIAL): Sequential Microwire Transfer

8.4.2.5.5 S_EN

- **Name:** Slave Enable Register
- **Description:** This register exists only when the SPI is configured as a master device. The register enables the individual slave select output lines from the SPI master.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 8-99 Slave Enable Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits.
1:0	SEL_EN	RW	0x0	Slave Select Enable. Each bit in this register corresponds to a slave select line (CSn) from the SPI master. **Value:** • 0x0 (NOT_SELECTED): No slave selected • 0x1 (CS0 SELECTED): Slave0 is selected • 0x2 (CS0 SELECTED): Slave1 isselected • 0x3 (CS0 and CS1 SELECTED): Both Slave0 and Slave1 are selected. **Note:** When the SPI is configured as a slave, this register serves no purpose.

8.4.2.5.6 BAUD

- **Name:** Baud Rate Register
- **Description:** This register exists only when the SPI is configured as a master device. The register derives the frequency of the serial clock that regulates the data transfer. It is impossible to write to this register when the SPI is enabled.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 8-100 Baud Rate Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits.
15:0	CLK_DIV	RW	0x0	SPI Clock Divider. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out = Fssi_clk/SCKDV **Note:** When the SPI is configured as a slave, this register serves no purpose.

8.4.2.5.7 TX_FIFO_TL

- **Name:** Transmit FIFO Threshold Level Register
- **Description:** This register controls the threshold value for the transmit FIFO memory.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 8-101 Transmit FIFO Threshold Level

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits.
2:0	TX_FIFO_THD	RW	0x0	Transmit FIFO Threshold. Controls the level of entries at (or below) which the transmit FIFO controller triggers an interrupt. The FIFO depth is configurable in 16 words; this register is sized to the number of address bits needed to access the FIFO. If users attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

8.4.2.5.8 RX_FIFO_TL

- **Name:** Receive FIFO Threshold Level
- **Description:** This register controls the threshold value for the receive FIFO memory.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x1C
- **Reset Value:** 0x00000000

Table 8-102 Receive FIFO Threshold Level

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits.
2:0	RX_FIFO_THD	RW	0x0	Receive FIFO Threshold. Controls the level of entries at (or above) which the receive FIFO controller triggers an interrupt. The FIFO depth is configurable in 16 words. This register is sized to the number of address bits needed to access the FIFO. If users attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.

8.4.2.5.9 TX_FIFO_LV

- **Name:** Transmit FIFO Level Register
- **Description:** This register contains the number of valid data entries in the transmit FIFO memory.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 8-103 Transmit FIFO Level Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits.
3:0	TX_FIFO_LV	R	0x0	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.

8.4.2.5.10 RX_FIFO_LV

- **Name:** Receive FIFO Level Register

- **Description:** This register contains the number of valid data entries in the receive FIFO memory. This register can be ready at any time.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 8-104 Receive FIFO Level Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits.
3:0	RX_FIFO_LV	R	0x0	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.

8.4.2.5.11 STAT

- **Name:** Status Register
- **Description:** This is a read-only register used to indicate the current transfer status, FIFO status, and any transmission/reception errors that may have occurred. The status register may be read at any time.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x28
- **Reset Value:** 0x00000006

Table 8-105 Status Register

Bits	Field Name	RW	Reset	Description
31:7	RSVD	R		Reserved bits.
6	DATA_COLN_ERR	R	0x0	Data Collision Error. This bit will be set if MISO input is asserted by other master, when the SPI master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. **Note:** When the SPI is configured as a slave, this register serves no purpose.
5	TX_ERR	R	0x0	Transmission Error. Set if the transmit FIFO is empty when a transfer is started. Data from the previous transmission is resent on the txd line. This bit is cleared when read.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (NO_ERROR): No Error • 0x1 (TX_ERROR): Transmission Error **Note:** When the SPI is configured as a master, this register serves no purpose.
4	RX_FIFO_FULL	R	0x0	Receive FIFO 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. **Value:** • 0x0 (NOT_FULL): Receive FIFO is not full • 0x1 (FULL): Receive FIFO is full
3	RX_FIFO_NE	R	0x0	Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. **Value:** • 0x0 (EMPTY): Receive FIFO is empty • 0x1 (NOT_EMPTY): Receive FIFO is not empty
2	TX_FIFO_EMPTY	R	0x1	Transmit FIFO Empty. When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. **Value:** • 0x0 (NOT_EMPTY): Transmit FIFO is not empty • 0x1 (EMPTY): Transmit FIFO is empty
1	TX_FIFO_NF	R	0x1	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. **Value:** • 0x0 (FULL): Transmit FIFO is full • 0x1 (NOT_FULL): Transmit FIFO is not Full

Bits	Field Name	RW	Reset	Description
0	BUSY	R	0x0	SSI Busy Flag. When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled. **Value:** • 0x0 (INACTIVE): SPI is idle or disabled • 0x1 (ACTIVE): SPI is actively transferring data

8.4.2.5.12 INT_MASK

- **Name:** Interrupt Mask Register
- **Description:** This read/write register masks or enables all interrupts generated by the SPI. When the SPI is configured as a slave device, the MSTIM bit field is not present. This changes the reset value from 0x3F for serial-master configurations to 0x1F for serial-slave configurations.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x2C
- **Reset Value:** master ? 0x0000003F : 0x0000001F

Table 8-106 Interrupt Mask Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits.
5	MULTI_M_CIM	RW	0x1	Multi-Master Contention Interrupt Mask. **Value:** • 0x0 (MASKED): Disable mst_intr interrupt • 0x1 (UNMASKED): Enable mst_intr interrupt **Note:** When the SPI is configured as a slave, this register serves no purpose.
4	RX_FIFO_FIM	RW	0x1	Receive FIFO Full Interrupt Mask **Value:** • 0x0 (MASKED): Disable rxf_intr interrupt • 0x1 (UNMASKED): Enable rxf_intr interrupt
3	RX_FIFO_OIM	RW	0x1	Receive FIFO Overflow Interrupt Mask **Value:** • 0x0 (MASKED): Disable rxo_intr interrupt • 0x1 (UNMASKED): Enable rxo_intr interrupt

Bits	Field Name	RW	Reset	Description
2	RX_FIFO_UIM	RW	0x1	Receive FIFO Underflow Interrupt Mask **Value:** • 0x0 (MASKED): Disable rxu_intr interrupt • 0x1 (UNMASKED): Enable rxu_intr interrupt
1	TX_FIFO_OIM	RW	0x1	Transmit FIFO Overflow Interrupt Mask **Value:** • 0x0 (MASKED): Disable txo_intr interrupt • 0x1 (UNMASKED): Enable txo_intr interrupt
0	TX_FIFO_EIM	RW	0x1	Transmit FIFO Empty Interrupt Mask **Value:** • 0x0 (MASKED): Disable txe_intr interrupt • 0x1 (UNMASKED): Enable txe_intr interrupt

8.4.2.5.13 INT_STAT

- **Name:** Interrupt Status Register
- **Description:** This register reports the status of the SPI interrupts after they have been enabled.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x30
- **Reset Value:** 0x00000000

Table 8-107 Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits.
5	MULTI_M_CIS	R	0x0	Multi-Master Contention Interrupt Status **Value:** • 0x0 (INACTIVE): mst_intr interrupt is not active after being enabled • 0x1 (ACTIVE): mst_intr interrupt is active after being enabled
4	RX_FIFO_FIS	R	0x0	Receive FIFO Full Interrupt Status **Value:** • 0x0 (INACTIVE): rxf_intr interrupt is not active after being enabled • 0x1 (ACTIVE): rxf_intr interrupt is active after being enabled

Bits	Field Name	RW	Reset	Description
3	RX_FIFO_OIS	R	0x0	Receive FIFO Overflow Interrupt Status **Value:** • 0x0 (INACTIVE): rxo_intr interrupt is not active after being enabled • 0x1 (ACTIVE): rxo_intr interrupt is active after being enabled
2	RX_FIFO_UIS	R	0x0	Receive FIFO Underflow Interrupt Status **Value:** • 0x0 (INACTIVE): rxu_intr interrupt is not active after being enabled • 0x1 (ACTIVE): rxu_intr interrupt is active after being enabled
1	TX_FIFO_OIS	R	0x0	Transmit FIFO Overflow Interrupt Status **Value:** • 0x0 (INACTIVE): txo_intr interrupt is not active after being enabled • 0x1 (ACTIVE): txo_intr interrupt is active after being enabled
0	TX_FIFO_EIS	R	0x0	Transmit FIFO Empty Interrupt Status **Value:** • 0x0 (INACTIVE): txe_intr interrupt is not active after being enabled • 0x1 (ACTIVE): txe_intr interrupt is active after being enabled

8.4.2.5.14 RAW_INT_STAT

- **Name:** Raw Interrupt Status Register
- **Description:** This read-only register reports the status of the SPI interrupts prior to being enabled.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x34
- **Reset Value:** 0x00000000

Table 8-108 Raw Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits.
5	MULTI_M_CRIS	R	0x0	Multi-Master Contention Raw Interrupt Status

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (INACTIVE): mst_intr interrupt is not active prior to being enabled • 0x1 (ACTIVE): mst_intr interrupt is active prior to being enabled
4	RX_FIFO_FRIS	R	0x0	Receive FIFO Full Raw Interrupt Status **Value:** • 0x0 (INACTIVE): rxf_intr interrupt is not active prior to being enabled • 0x1 (ACTIVE): rxf_intr interrupt is active prior to being enabled
3	RX_FIFO_ORIS	R	0x0	Receive FIFO Overflow Raw Interrupt Status **Value:** • 0x1 (ACTIVE): rxo_intr interrupt is not active prior to being enabled • 0x0 (INACTIVE): rxo_intr interrupt is active prior being enabled
2	RX_FIFO_URIS	R	0x0	Receive FIFO Underflow Raw Interrupt Status **Value:** • 0x0 (INACTIVE): rxu_intr interrupt is not active prior to being enabled • 0x1 (ACTIVE): rxu_intr interrupt is active prior to being enabled
1	TX_FIFO_ORIS	R	0x0	Transmit FIFO Overflow Raw Interrupt Status **Value:** • 0x0 (INACTIVE): txo_intr interrupt is not active prior to being enabled • 0x1 (ACTIVE): txo_intr interrupt is active prior being enabled
0	TX_FIFO_ERIS	R	0x0	Transmit FIFO Empty Raw Interrupt Status **Value:** • 0x0 (INACTIVE): txe_intr interrupt is not active prior to being enabled • 0x1 (ACTIVE): txe_intr interrupt is active prior being enabled

8.4.2.5.15 TX_FIFO_OIC

- **Name:** Transmit FIFO Overflow Interrupt Clear Register
- **Description:** Transmit FIFO Overflow Interrupt Clear Register.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 8-109 Transmit FIFO Overflow Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits.
0	TX_FIFO_OIC	R	0x0	Clear Transmit FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the txo_intr interrupt; writing has no effect.

8.4.2.5.16 RX_FIFO_OIC

- **Name:** Receive FIFO Overflow Interrupt Clear Register
- **Description:** Receive FIFO Overflow Interrupt Clear Register.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x3C
- **Reset Value:** 0x00000000

Table 8-110 Receive FIFO Overflow Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits.
0	RX_FIFO_OIC	R	0x0	Clear Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the rxo_intr interrupt; writing has no effect.

8.4.2.5.17 RX_FIFO_UIC

- **Name:** Receive FIFO Underflow Interrupt Clear Register
- **Description:** Receive FIFO Underflow Interrupt Clear Register.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x40

- **Reset Value:** 0x00000000

Table 8-111 Receive FIFO Underflow Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits.
0	RX_FIFO_UIC	R	0x0	Clear Receive FIFO Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the rxu_intr interrupt; writing has no effect.

8.4.2.5.18 MULTI_M_IC

- **Name:** Multi-Master Interrupt Clear Register
- **Description:** Multi-Master Interrupt Clear Register.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x44
- **Reset Value:** 0x00000000

Table 8-112 Multi-Master Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits.
0	MULTI_M_IC	R	0x0	Clear Multi-Master Contention Interrupt. This register reflects the status of the interrupt. A read from this register clears the mst_intr interrupt; writing has no effect.

8.4.2.5.19 INT_CLR

- **Name:** Interrupt Clear Register
- **Description:** Interrupt Clear Register.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x48
- **Reset Value:** 0x00000000

Table 8-113 Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits.

Bits	Field Name	RW	Reset	Description
0	INT_CLR	R	0x0	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the txo_intr, rxu_intr, rxo_intr, and the mst_intr interrupts. Writing to this register has no effect.

8.4.2.5.20 DMA_CTRL

- **Name:** DMA Control Register
- **Description:** The register is used to enable the DMA Controller interface operation.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x4C
- **Reset Value:** 0x00000000

Table 8-114 DMA Control Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits.
1	TX_DMA_EN	RW	0x0	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. **Value:** • 0x0 (DISABLE): Transmit DMA disabled • 0x1 (ENABLED): Transmit DMA enabled
0	RX_DMA_EN	RW	0x0	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel **Value:** • 0x0 (DISABLE): Receive DMA disabled • 0x1 (ENABLED): Receive DMA enabled

8.4.2.5.21 DMA_TX_DL

- **Name:** DMA Transmit Data Level
- **Description:** This register controls the threshold value for the transmit FIFO memory.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 8-115 DMA Transmit Data Level

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits.
2:0	DMA_TX_DL	RW	0x0	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, 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, and TX_DMA_EN = 1.

8.4.2.5.22 DMA_RX_DL

- **Name:** DMA Receive Data Level
- **Description:** This register controls the threshold value for the receive FIFO memory.
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x54
- **Reset Value:** 0x00000000

Table 8-116 DMA Receive Data Level

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	DMA_RX_DL	RW	0x0	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMA_RX_DL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RX_DMA_EN=1.

8.4.2.5.23 DATA

- **Name:** Data Register
- **Description:** The SPI data register is a 32-bit read/write buffer for the transmit/receive FIFOs. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data is moved into the transmit FIFO buffer; a write can occur only when SSI_EN[0] = 1. FIFOs are reset when SSI_EN[0] = 0
- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0x60
- **Reset Value:** 0x00000000

Table 8-117 Data Register

Bits	Field Name	RW	Reset	Description
31:0	DATA	RW	0x0	Data Register. When writing to this register, you must right-justify the data. Read data is automatically right-justified.

8.4.2.5.24 RX_SMP_DLY

Name: Receive sample delay Register

Description: This register control the number of SCLK cycles that are delayed (from the default sample time) before the actual sample of the rxd input occurs. It is impossible to write to this register when the SPI is enabled.

- **Base Address:** 0x4000C000 + x*0x100 (x: 0 for master, 1 for slave)
- **Offset:** 0xF0
- **Reset Value:** 0x00000000

Table 8-118 Data Register

Bits	Field Name	RW	Reset	Description
3:0	RX_SMP_DLY	RW	0x0	Receive sample delay

8.4.2.6 Electrical Specifications

8.4.2.6.1 SPIM Electrical Specifications

Table 8-119 SPIM electrical specifications

Parameter	Description	Min.	Typ.	Max.	Unit
f_{SCK}	Clock frequency			48	MHz
t_{CSCK}	SCK clock period	20.83			ns
t_{RSCK}	SCK rise time, 15 pF loading			$t_{RF,15pF}$	ns
t_{FSCK}	SCK fall time, 15 pF loading			$t_{RF,15pF}$[2]	ns
t_{WSCKH}	SCK clock high time	7.625			ns
t_{WSCKL}	SCK clock low time	7.625[4]			ns
t_{SUMI}	Data input setup time (MISO to SCK edge)	5			ns
t_{HMI}	Data input hold time (SCK edge to MISO change)	5			ns
t_{VMO}	Data output valid time (SCK edge to MOSI valid)	0			ns
t_{HMO}	MOSI hold time after CLK edge			20	ns

[1] *The SCK rise time and the SCK fall time are obtained with SPIM timing shown in Figure 8-21.*

[2] *The SCK rise time and the SCK fall time are obtained with SPIM timing shown in #unique_380/unique_380_Connect_42_d6e362.*

[3] *The SCK clock high time and SCK clock low time are calculated by ($t_{CSCK}/2$)–t_{RSCK} and ($t_{CSCK}/2$)–t_{FSCK}, respectively.*

[4] *The SCK clock high time and SCK clock low time are calculated by ($t_{CSCK}/2$)–t_{RSCK} and ($t_{CSCK}/2$)–t_{FSCK}, respectively.*

[1]The SCK rise time and the SCK fall time are obtained with SPIM timing shown in Figure 8-21.
[3]The SCK clock high time and SCK clock low time are calculated by (t_{CSCK}/2)–t_{RSCK} and (t_{CSCK}/2)–t_{FSCK}, respectively.

The SPIM timing diagram is as below:

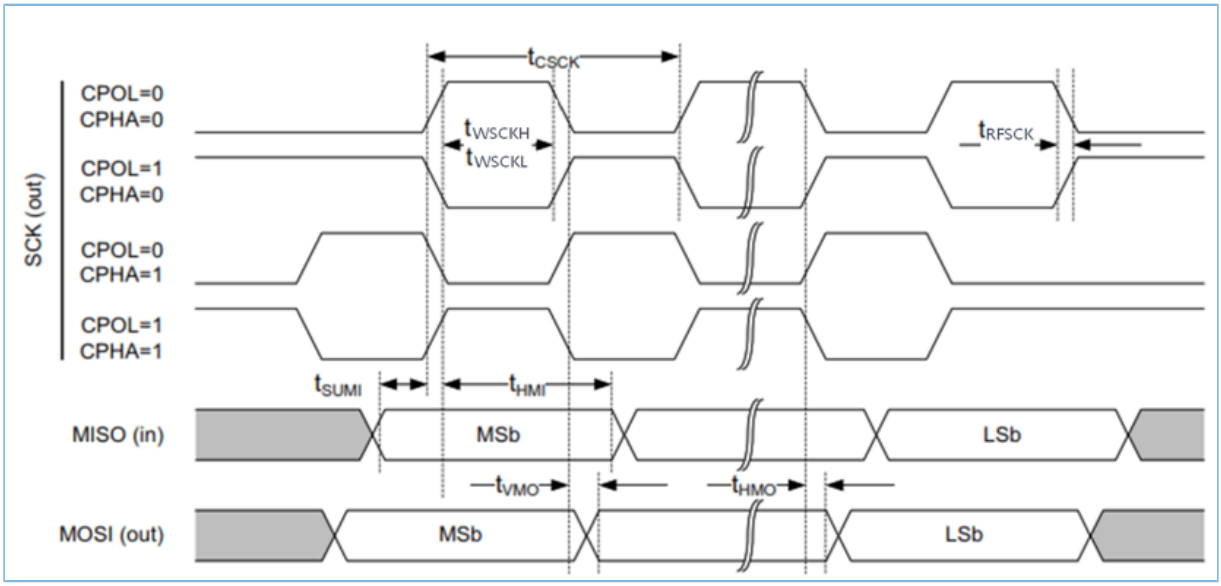

Figure 8-21 SPIM timing diagram

8.4.2.6.2 SPIS Electrical Specifications

Table 8-120 SPIS electrical specifications

Parameter	Description	Min.	Typ.	Max.	Unit
f_{SCK}	Clock frequency			16	MHz
t_{CSCK}	SCK input clock period	62.5			ns
t_{RFSCK}	SCK input clock rise/fall time			22	ns
t_{WSCKH}	SCK input clock high time	22			ns
t_{WSCKL}	SCK input clock low time	22			ns
t_{SUCSN}	CSN input setup time	62.5			ns
t_{HCSN}	CSN input hold time	31.25			ns
t_{ASO}	Data output access time (from CSN to MISO valid)			62.5	ns
t_{DISSO}	Data output disable time (from CSN to MISO disabled)			62.5	ns
t_{VSO}	Data output valid time (from SCK to MISO valid)			20	ns
t_{HSO}	Data output hold time (from SCK to MISO change)	8			ns
t_{SUSI}	Data input setup time (from MOSI to SCK)	18			ns
t_{HSI}	Data input hold time (from SCK to MOSI change)	0			ns

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

Note:

If SPIS performs RX and TX concurrently, the maximum bit rate decreases to 8 Mbps.

The SPIS timing diagram is shown below:

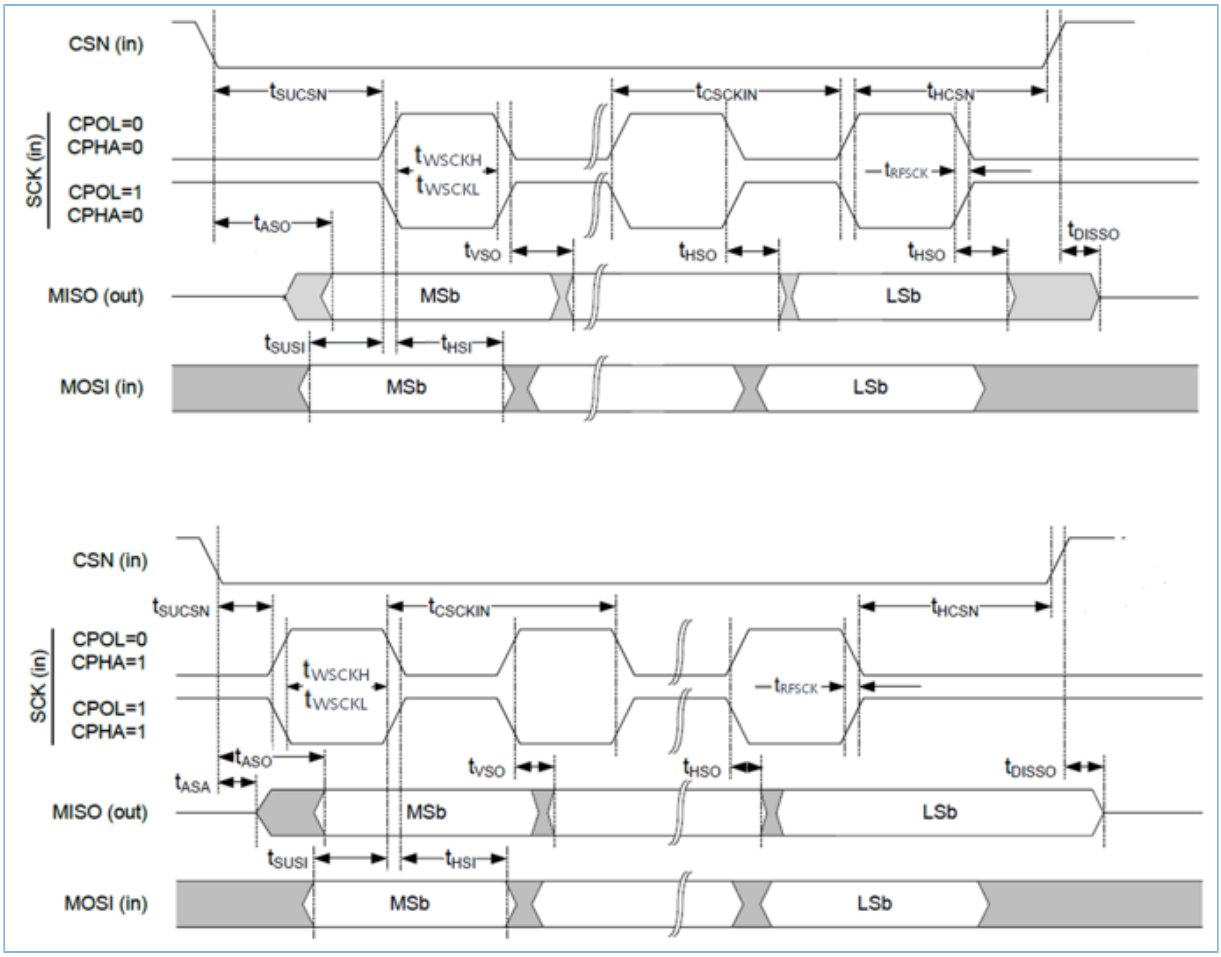

Figure 8-22 SPIS timing diagram

8.4.3 I2C

8.4.3.1 Introduction

The Inter-Integrated Circuit (I2C) bus is a two-wire serial interface that is widely used for low-speed communication between chipsets in a system. Using I2C, a microcontroller can communicate with peripherals such as sensors, data converters, and I/O interfaces.

GR5526 SoC has six I2C module instances: I2C0, I2C1, I2C2, I2C3, I2C4, and I2C5 that can be configured as either Master or Slave.

8.4.3.2 Main Features

The two-wire I2C serial interface consists of a serial data line (SDA) and a serial clock line (SCL) that support:

- Standard Mode (up to 100 kbit/s)
- Fast Mode (up to 400 kbit/s)
- Fast Plus Mode (up to 1000 kbit/s)
- High-speed Mode (up to 3.4 Mbit/s)
- Clock synchronization
- Master or Slave I2C operation
- 7-bit or 10-bit addressing
- Mixed 7-bit and mixed 10-bit format transfer
- Transmit and receive FIFO depth at 32 bytes
- Interrupt or polled-mode operation
- Simple software interface
- DMA support

8.4.3.3 Functional Description

The I2C Controller is made up of an APB slave interface, an I2C interface, and FIFO logic to maintain consistency between the two interfaces, as well as DMA and interrupt logics. A simplified block diagram is illustrated in Figure 8-23.

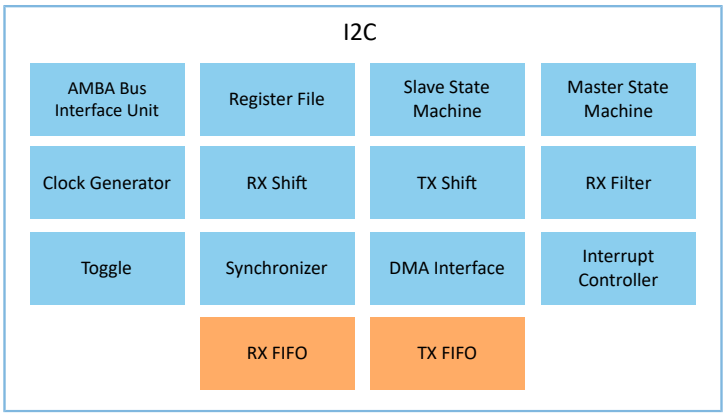

Figure 8-23 I2C block diagram

The following list describes each of the major blocks shown above.

- AMBA Bus Interface Unit connects to APB bus.
- Register File contains configuration registers and is the interface with software.
- Slave State Machine follows the protocol for a slave and monitors bus for address match.
- Master State Machine generates the I2C protocol for the master transfers.
- Clock Generator produces the required timing to do the following:

- Generate the SCL clock when configured as a master.
- Check for bus idle.
- Generate a START and a STOP.
- Set up the data and hold the data.

- RX Shift takes data into the design and extracts it in byte format.
- TX Shift presents data supplied by GPU for transfer on the I2C bus.
- RX Filter detects the events in the bus; for example, start, stop, and arbitration lost.
- Toggle generates pulses on both sides and toggles to transfer signals across clock domains.
- Synchronizer transfers signals from one clock domain to another.
- DMA Interface generates handshaking signals to the central DMA controller in order to automate the data transfer without CPU intervention.
- Interrupt Controller generates raw interrupt and interrupt flags, allowing them to be set and cleared.
- RX FIFO/TX FIFO holds the RX FIFO and TX FIFO register banks and controllers, along with their status levels.

8.4.3.3.1 I2C Operation

The I2C is a synchronous serial interface. The SDA line is a bidirectional signal and changes only when the SCL line is low, except for STOP, START, and RESTART conditions. The output drivers are open-drain or open-collector to perform wired-AND functions on the bus. The maximum number of devices on the bus is limited by only the maximum capacitance specification of 400 pF. Data is transmitted in byte packages.

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 acknowledgement of data is sent by the device that is receiving data, which can be either a master or a slave. The I2C protocol also allows multiple masters to reside on the I2C bus and uses an arbitration procedure to determine the bus ownership.

When a master wants to communicate with a slave, the master transmits a START/RESTART condition that is then followed by the slave address and a control bit ($R/\overline{W}$) to determine if the master wants to transmit data to or receive data from the slave. The slave then sends an acknowledge (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. This behavior is illustrated in Figure 8-24.

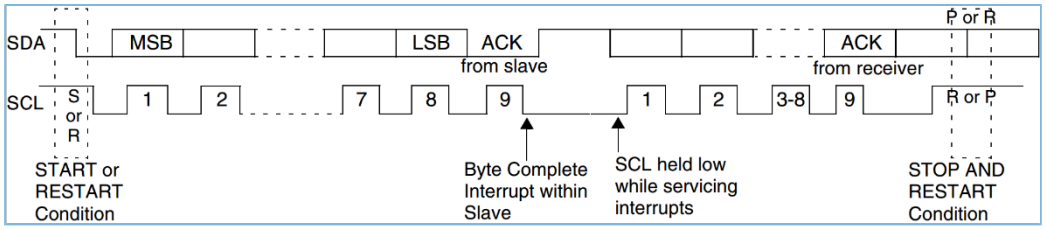

Figure 8-24 Data transfer on the I2C Bus

Note that, I2C block should only be set to operate as an I2C master or an I2C slave, but not both simultaneously. This is achieved by ensuring that bit 0 and bit 6 of the I2C Control Register (CTRL) are never set to 0 and 1, respectively.

8.4.3.3.2 START and STOP Conditions

When I2C operates as a master (writing 1 to CTRL[0], and writing 1 to CTRL[6]), putting data into the transmit FIFO causes the I2C block to generate a START condition on the I2C bus. Writing 1 to DATA_CMD[9] causes the I2C block to generate a STOP condition on the I2C bus; a STOP condition is not issued if this bit is not set, even if the transmit FIFO is empty. This behavior is illustrated as follows.

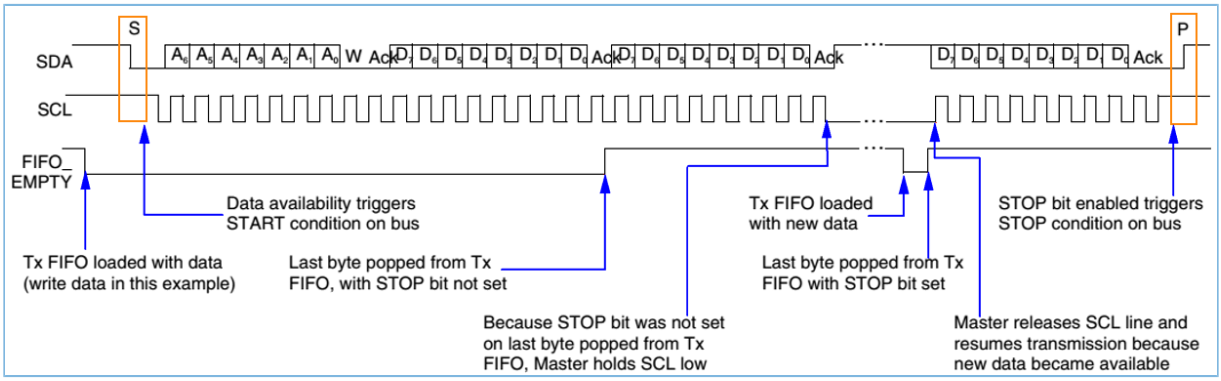

Figure 8-25 STOP generation

When operating as a slave (writing 0 to CTRL[0], and writing 0 to CTRL[6]), I2C block does not generate START and STOP conditions, as per the protocol. However, if a read request is made to the I2C block, it holds the SCL line low until read data has been supplied to it. This stalls the I2C bus until read data is provided to the slave I2C block, or the I2C block slave is disabled by writing 0 to I2C ENABLE Register (EN[0]). When the bus is idle, both the SCL and SDA signals are pulled high through external pull-up resistors on the bus. When a master wants to start a transmission on the bus, the master issues a START condition. This is defined as a high-to-low transition of the SDA signal while SCL is high. When a master wants to terminate the transmission, the master issues a STOP condition. This is defined as a low-to-high transition of the SDA line while SCL is high. Figure 8-26 shows the timing of the START and STOP conditions. When data is being transmitted on the bus, the SDA line must be stable when the SCL is high.

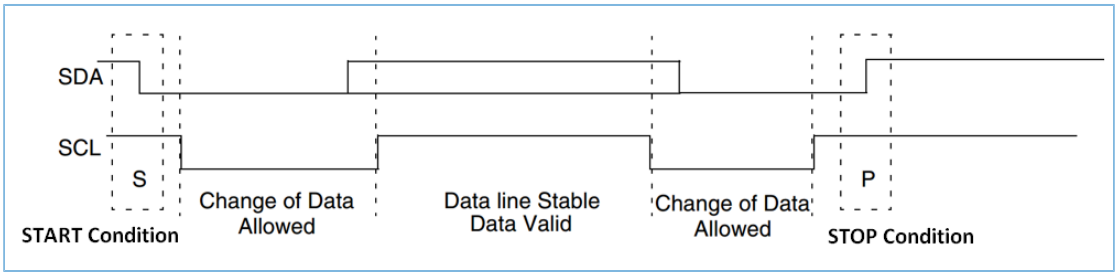

Figure 8-26 Timing of START and STOP conditions

8.4.3.3.3 Addressing Slave

There are two address formats: the 7-bit address format and the 10-bit address format. The I2C block supports mixed formats of read and write transactions in 7-bit or 10-bit addressing modes. The I2C block does not support mixed address or mixed address format, that is, a 7-bit address transaction followed by a 10-bit address transaction or vice versa (combined format transactions).

To initiate combined format transfers, CTRL[5] should be enabled. With this value set and operating as a master, when the I2C block completes an I2C transfer, it checks the transmit FIFO and executes the next transfer. If the direction of this transfer differs from the previous transfer, the combined format is used to issue the transfer. If the transmit FIFO is empty when the current I2C transfer completes, a STOP is issued and the next transfer is issued following a START condition.

7-bit Address Format

Using the 7-bit address format, users should write 0 to CTRL[3] and the slave address is stored at I2C Slave Address Register (S_ADDR[6:0]). The first seven bits (bits 7:1) of the first byte set the slave address and the least significant bit (bit 0) (LSB) is the R/W bit as shown in Figure 8-27. When bit 0 ($R/\overline{W}$) is set to 0, the master writes to the slave. When bit 0 ($R/\overline{W}$) is set to 1, the master reads from the slave.

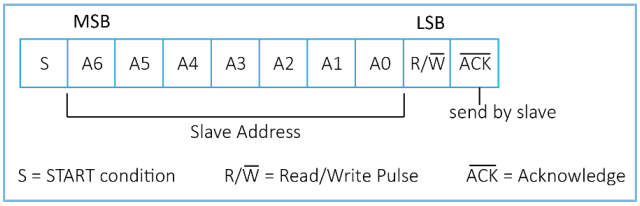

Figure 8-27 7-bit address format

10-bit Address Format

Using the 10-bit address format, users should write 1 to CTRL[3] and the slave address is stored at I2C Slave Address Register (S_ADDR[9:0]). Two bytes are transferred to set the 10-bit address. The transfer of the first byte contains the following bit definition. The first five bits (bits 7:3) notify the slave that this is a 10-bit transfer followed by the next two bits (bits 2:1), which set the slave address (bits 9:8), and the LSB bit (bit 0) is the $R/\overline{W}$ bit. The second byte transferred sets bits 7:0 of the slave address.

Figure 8-28 shows the 10-bit address format.

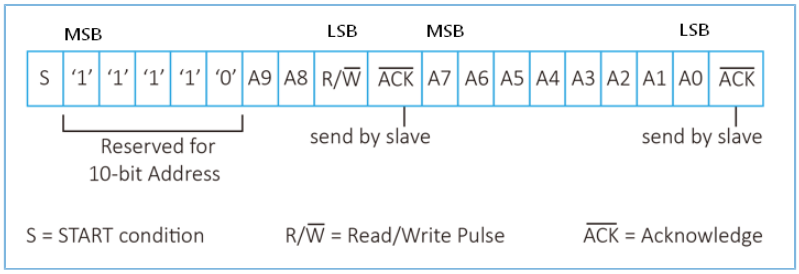

Figure 8-28 10-bit address format

Table 8-121 defines the special purpose and reserved first byte addresses.

Table 8-121 I2C definition of bits in first byte

Slave Address	R/W Bit	Description
0000 000	0	General Call Address. DW_apb_i2c places the data in the receive buffer and issues a General Call interrupt.
0000 000	1	START byte
0000 001	X	CBUS address. I2C ignores these accesses.
0000 010	X	Reserved
0000 011	X	High-speed master code
1111 1XX	X	Reserved
1111 0XX	X	10-bit slave addressing
0001 000	X	SMBus Host
0001 100	X	SMBus Alert Response Address
1100 001	X	SMBus Device Default Address

8.4.3.3.4 Transmitting and Receiving

A master can initiate data transmission and reception to/from the bus, acting as either a master-transmitter or master-receiver. A slave responds to requests from the master to either transmit data or receive data to/from the bus, acting as either a slave-transmitter or slave-receiver, respectively.

Master-Transmitter and Slave-Receiver

All data is transmitted in byte format, with no limit on the number of bytes transferred per data transfer. After the master sends the address and $R/\overline{W}$ bit or the master transmits a byte of data to the slave, the slave-receiver must respond with the ACK signal. When a slave-receiver does not respond with an ACK pulse, the master aborts the transfer by issuing a STOP condition. The slave must leave the SDA line high so that the master can abort the transfer.

If the master-transmitter is transmitting data as shown in Figure 8-29, the slave-receiver responds to the master-transmitter with an ACK pulse after every data byte is received.

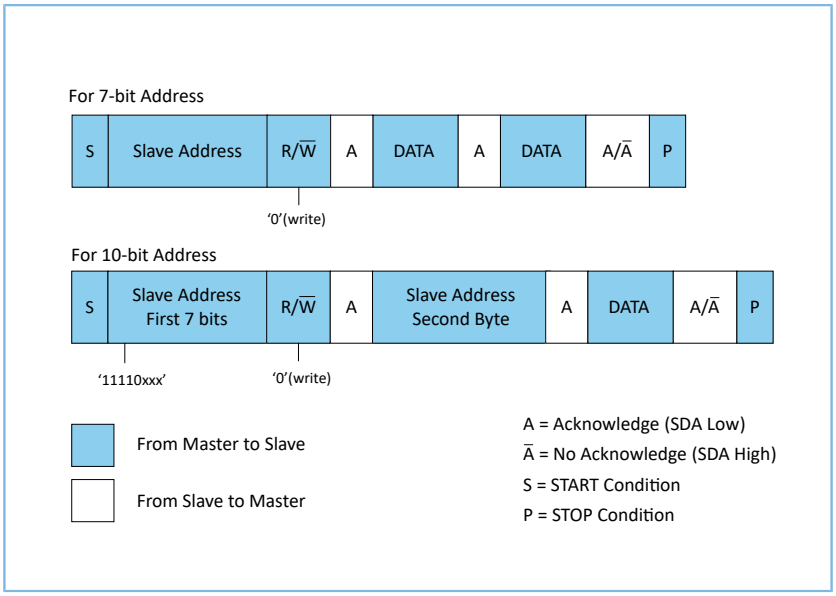

Figure 8-29 Master-transmitter protocol

Master-Receiver and Slave-Transmitter

If the master is receiving data as shown in Figure 8-30, the master responds to the slave-transmitter with an ACK pulse after a byte of data has been received, except for the last byte. This is the way in which the master-receiver notifies the slave-transmitter that this is the last byte. The slave-transmitter relinquishes the SDA line after detecting the No Acknowledge (NACK) so that the master can issue a STOP condition.

When a master does not want to relinquish the bus with a STOP condition, the master can issue a RESTART condition. This is identical to a START condition except it occurs after the ACK pulse. Operating in master mode, the I2C block can then communicate with the same slave using a transfer of a different direction.

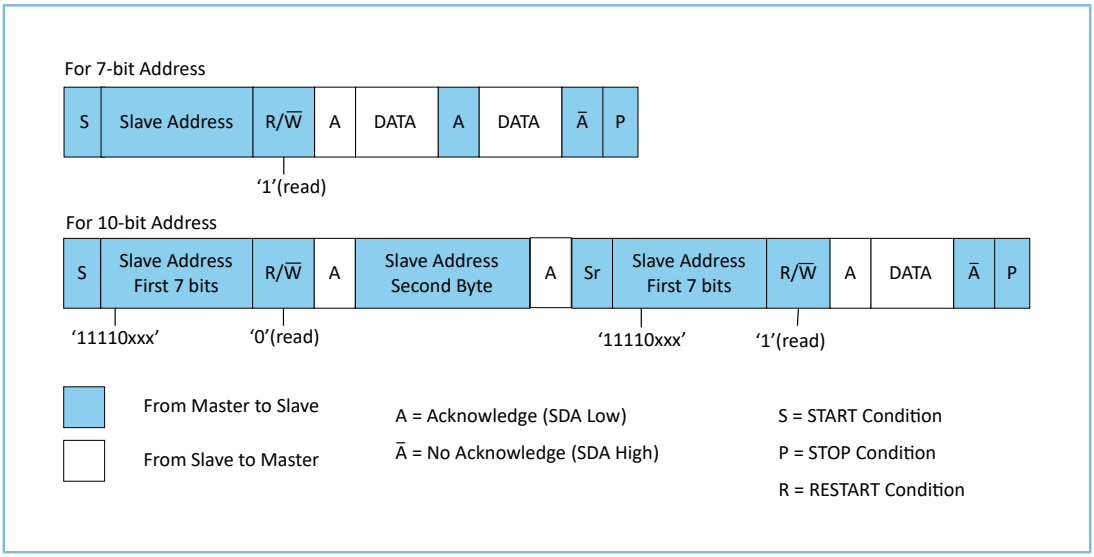

Figure 8-30 Master-receiver protocol

Bus Clear Feature

I2C block supports the bus clear feature that provides graceful recovery of data (SDA) and clock (SCL) lines during unlikely events in which either the clock or data line is stuck at low.

SDA Line Stuck at Low Recovery

In cast of SDA line stuck at low, master performs the following actions to recover as shown in Figure 8-31.

- Master sends a maximum of 9 clock pulses to recover the bus low within those 9 clocks.
  - The number of clock pulses will vary with the number of bits that remain to be sent by the slave. As the maximum number of bits is 9, master sends up to 9 clock pluses and allows the slave to recover it.
  - The master attempts to assert a Logic 1 on the SDA line and check whether SDA is recovered. If the SDA is not recovered, it will continue to send a maximum of 9 SCL clocks.
- If SDA line is recovered within 9 clock pulses, then the master will send the STOP to release the bus.
- If SDA line is not recovered even after 9th clock pulse, then the system needs a hardware reset.

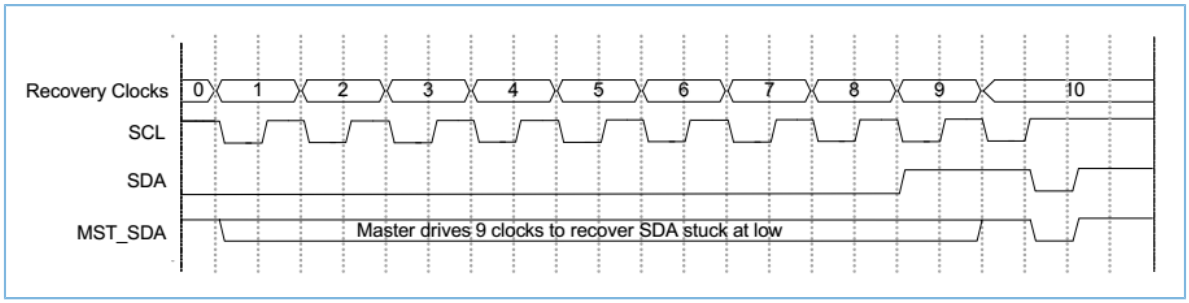

Figure 8-31 SDA recovery with 9 SCL clocks

The detailed flow to recover the SDA stuck at LOW is shown in Figure 8-32.

SCL Line Stuck at Low Recovery

In the unlikely event (due to an electric failure of a circuit) where the clock (SCL) is stuck at LOW, there is no effective method to overcome this problem but to reset the bus using the hardware reset signal. The detailed flow to recover the SCL stuck at LOW is shown in Figure 8-32.

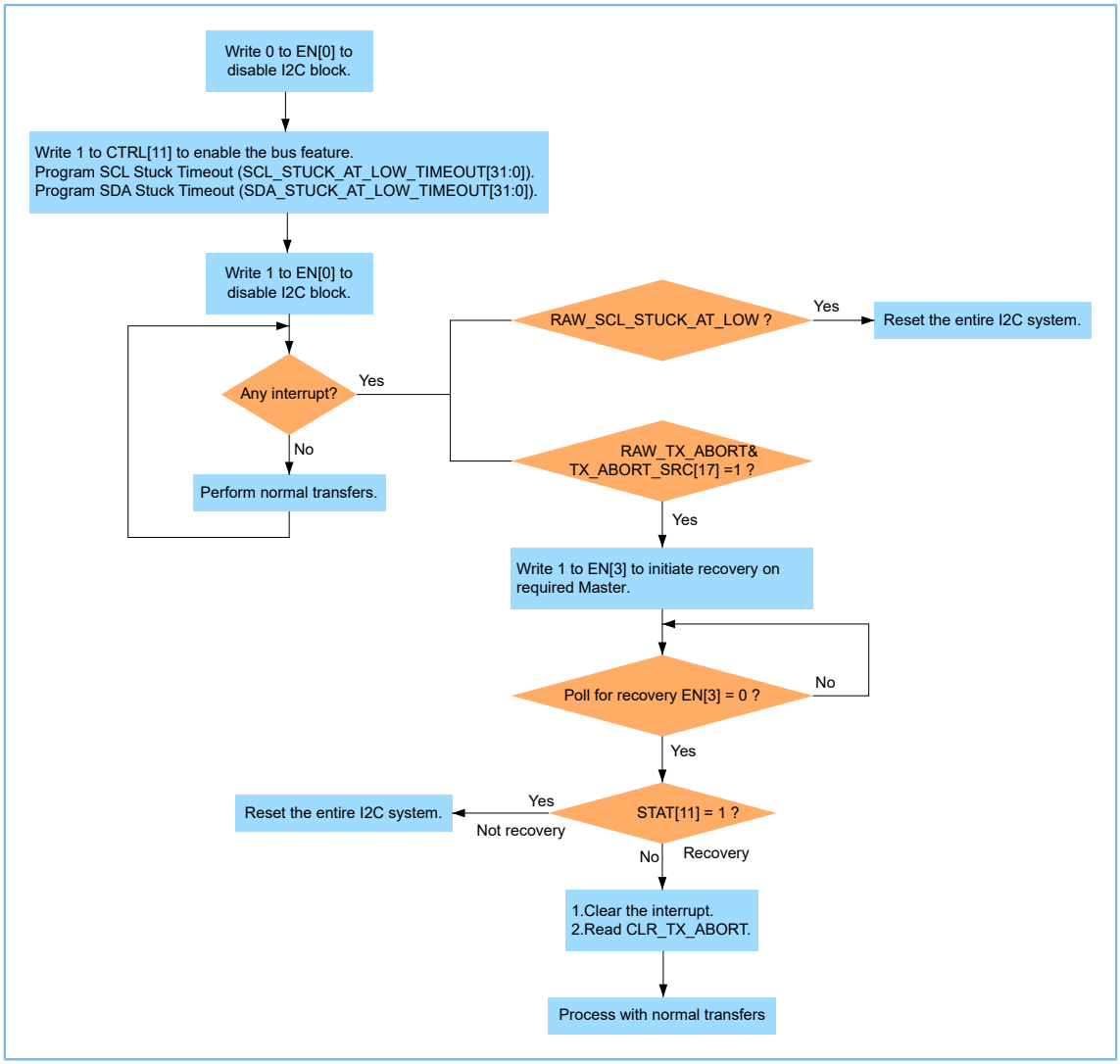

Figure 8-32 Flowchart for SCL and SDA bus recovery

8.4.3.3.5 SDA Hold Time

The I2C protocol specification requires 300 ns of hold time on the SDA signal (t_{HD_DAT}, SDA hold time, see Figure 8-35 in standard mode and fast mode, and a hold time long enough to bridge the undefined part between logic 1 and logic 0 of the falling edge of SCL in high speed mode and fast mode plus.

Board delays on the SCL and SDA signals can mean that the hold-time requirement is met at the I2C master, but not at the I2C slave (or vice versa). As each application encounters differing board delays, the I2C block contains a software programmable register (SDA_HOLD) to enable dynamic adjustment of the SDA hold time.

SDA_HOLD[15:0] are used to control the hold time of SDA during transmit in both slave and master mode (After SCL goes from HIGH to LOW).

SDA_HOLD[23:16] are used to extend the SDA transition (if any) whenever SCL is HIGH in the receiver (in either master or slave mode).

If different SDA hold times are required for different speed modes, the SDA_HOLD register must be reprogrammed when the speed mode is being changed. The SDA_HOLD register can be programmed only when the DW_apb_i2c is disabled (EN[0] = 0).

SDA Hold Timings in Receiver

When I2C block acts as a receiver, according to the I2C protocol, the device should internally hold the SDA line to bridge undefined gap between logic 1 and logic 0 of SCL.

SDA_HOLD[23:16] can be used to alter the internal hold time which I2C block applies to the incoming SDA line. Each value in the SDA_HOLD[23:16] represents a unit of one sclk (in serial clock implementation) period. The minimum value of SDA_HOLD[23:16] is 0. This hold time is applicable only when SCL is HIGH. The receiver does not extend the SDA after SCL goes LOW internally. Figure 8-33 shows the I2C block as receiver with SDA_HOLD[23:16] being programmed to greater than or equal to 3. If SDA_HOLD[23:16] are greater than 3, I2C block does not hold SDA beyond 3 sclk cycles, because SCL goes low internally.

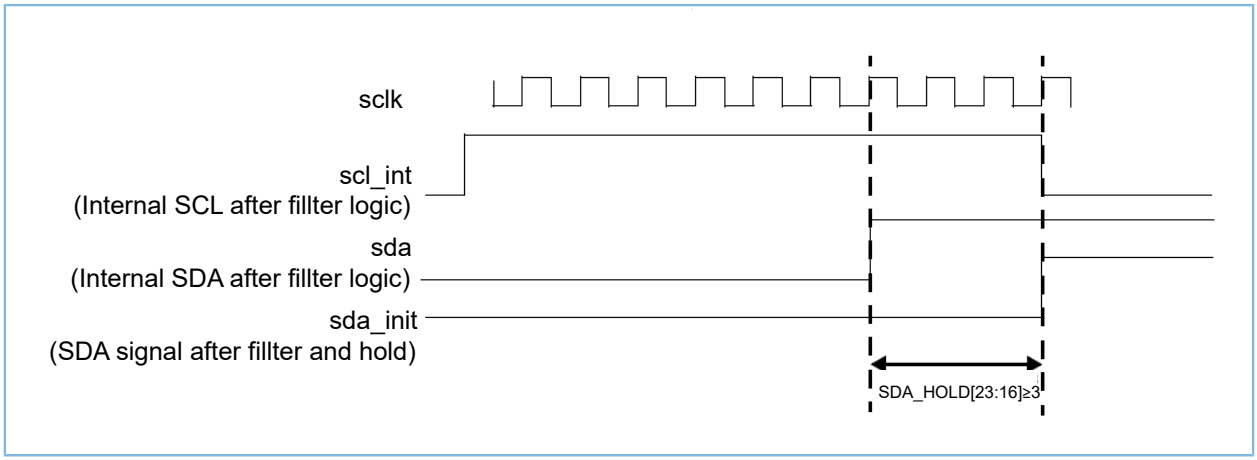

Figure 8-33 I2C block as receiver with SDA hold time

The maximum values of SDA_HOLD[23:16] that can be programmed in the register for the respective speed modes are derived from the equations shown in the table below.

Table 8-122 Maximum values for SDA_HOLD[23:16]

Speed Mode	Maximum SDA_HOLD[23:16] Value
Standard Mode	SS_CLK_HCNT[15:0] – FS_SPKLEN[7:0] -3
Fast Mode or Fast Mode Plus	FS_CLK_HCNT[15:0] – FS_SPKLEN[7:0] -3
High Speed	Min{FS_CLK_HCNT[15:0] – FS_SPKLEN[7:0] -3, HS_CLK_HCNT[15:0] – HS_SPKLEN[7:0] -3}

SDA Hold Timings in Transmitter

The SDA_HOLD [15:0] can be used to alter the timing of the generated SDA (data_oe, see Figure 8-34) signal by the I2C block. Each value in the SDA_HOLD [15:0] represents a unit of one sclk period.

When the I2C block is operating in master mode, the minimum t_{HD_DAT} timing is one ic_clk period. Therefore even when SDA_TX_HOLD has a value of zero, the I2C block drives SDA (data_oe) one sclk cycle after driving SCL (scl_oe)

to logic 0. For all other values of SDA_HOLD[15:0], drive on SDA (data_oe) occurs SDA_HOLD[15:0] sclk cycles after driving SCL (scl_oe) to logic 0.

When the I2C block is operating in slave mode, the minimum t_{HD_DAT} timing is SPKLEN + 7 sclk periods, where SPKLEN is FS_SPKLEN[7:0] (operating in standard mode, fast mode, or fast mode plus) or HS_SPKLEN[7:0] (operating in high speed mode).

This delay allows for synchronization and spike suppression on the SCL sample. Therefore, even when SDA_HOLD[15:0] has a value less than SPKLEN + 7, the I2C block drives SDA (data_oe) SPKLEN + 7 ic_clk cycles after SCL (scl_oe) has transitioned to logic 0. For all other values of SDA_HOLD[15:0], drive on SDA occurs SDA_HOLD[15:0] sclk cycles after SCL has transitioned to logic 0.

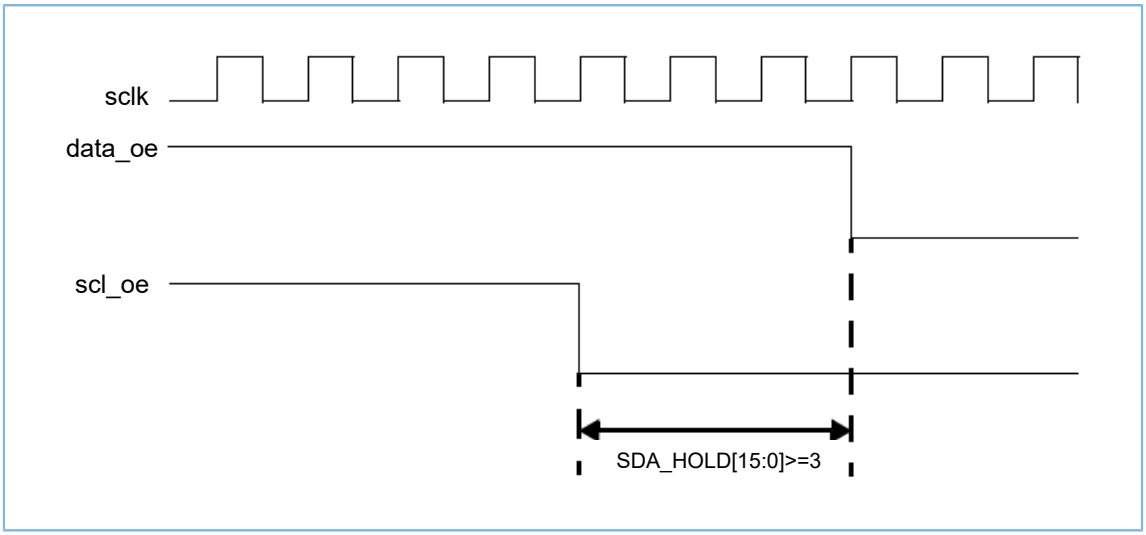

Figure 8-34 I2C Master implementing t_{HD_DAT} with SDA_HOLD[15:0] = 3

When the I2C block is operating in Slave Mode, the minimum t_{HD_DAT} timing is SPKLEN + 7 sclk periods, where SPKLEN is FS_SPKLEN[7:0] (operating in standard mode, fast mode, or fast mode plus) or HS_SPKLEN[7:0] (operating in high speed mode).

8.4.3.4 Registers

8.4.3.4.1 CTRL

- **Name**: I2C Control Register
- **Description**: This register can be written only when the I2C is disabled, which corresponds to the EN[0] register being set to 0. Writes at other times have no effect.
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x00
- **Reset Value**: 0x0000007F

Table 8-123 I2C Control Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R		Reserved bits
11	BUS_CLEAR_FEATURE_CTRL	RW	0x0	In Master mode **Value**: • 0x1: Bus Clear Feature is enabled. • 0x0: Bus Clear Feature is disabled.
10	STOP_DET_M_ACTIVE	RW	0x0	In Master mode **Value**: • 0x1: Issues the STOP_DET interrupt only when master is active. • 0x0: Issues the STOP_DET irrespective of whether master is active or not.
9	RX_FIFO_FULL_HLD_CTRL	RW	0x0	This bit controls whether I2C should hold the bus when the RX FIFO is physically full. **Value**: • 0x1 (ENABLED): Hold bus when RX_FIFO is full. • 0x0 (DISABLED): Overflow when RX_FIFO is full.
8	TX_EMPTY_CTRL	RW	0x0	This bit controls the generation of the TX_EMPTY interrupt, as described in the RAW_INT_STAT register. **Value**: • 0x0 (DISABLED): Default behavior of TX_EMPTY interrupt. • 0x1 (ENABLED): Controlled generation of TX_EMPTY interrupt.
7	STOP_DET_INT	RW	0x0	In slave mode **Value**: • 0x0: Issues the STOP_DET irrespective of whether it is addressed or not. • 0x1: Issues the STOP_DET interrupt only when it is addressed. **Note**: During a general call address, the slave does not issue the STOP_DET interrupt if STOP_DET_INT = 1, even if the slave responds to the general call address by generating ACK. The STOP_DET interrupt is generated only when the transmitted address matches the slave address (S_ADDR).

Bits	Field Name	RW	Reset	Description
6	S_DIS	RW	0x1	This bit controls whether I2C has its slave disabled. By default, the slave is always disabled (in reset state as well). If you need to enable it after reset, set this bit to 0. If this bit is set (slave is disabled), I2C functions only as a master and does not perform any action that requires a slave. **Note**: Software should ensure that if this bit is written with 0, then bit 0 should also be written with 0. **Value**: • 0x0 (S_EN): Slave mode is enabled. • 0x1 (S_DIS): Slave mode is disabled.
5	RESTART_EN	RW	0x1	Determine whether RESTART conditions may be sent when I2C acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in several I2C operations. When RESTART is disabled, the master is prohibited from performing the following functions: • Performing any high-speed mode operation. • Performing direction changes in combined format mode. • Performing a read operation with a 10-bit address. By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple I2C transfers. If the above operations are performed, it will result in setting bit 6 (TX_ABORT) of the RAW_INT_STAT register. **Value**: • 0x0 (DISABLED): Master restart disabled. • 0x1 (ENABLED): Master restart enabled.
4	ADDR_BIT_M	RW	0x1	This bit controls whether the I2C starts its transfers in 7-bit or 10-bit addressing mode when I2C acts as a master. **Value**: • 0x0 (ADDR_7BITS): Master 7-bit addressing mode • 0x1 (ADDR_10BITS): Master 10-bit addressing mode

Bits	Field Name	RW	Reset	Description
3	ADDR_BIT_S	RW	0x1	When I2C acts as a slave, this bit controls whether the I2C responds to 7- or 10-bit addresses. **Value**: • 0x0 (ADDR_7BITS): Slave 7-Bit addressing • 0x1 (ADDR_10BITS): Slave 10-Bit addressing
2:1	SPEED	RW	0x3	These bits control at which speed the I2C operates; its setting is relevant only if one is operating the I2C in master mode. Hardware protects against illegal values being programmed by software. These bits must be programmed appropriately for slave mode, as it is used to capture correct value of spike filter as per the speed mode. This register should be programmed only with a value in the range of 1 to 3. • 0x1: Standard Mode (100 kbit/s) • 0x2: Fast Mode (≤ 400 kbit/s) or Fast Plus Mode (≤ 1000 kbit/s) • 0x3: High-speed Mode (2 Mbit/s) **Value**: • 0x1 (STANDARD): Standard Speed Mode of operation • 0x2 (FAST): Fast or Fast Plus Mode of operation • 0x3 (HIGH): High-speed Mode of operation
0	M_MODE	RW	0x1	This bit controls whether the I2C master is enabled. **Note**: Software should ensure that if this bit is written with '1', then bit 6 should also be written with '1'. **Value**: • 0x0 (DISABLED): Master mode is disabled. • 0x1 (ENABLED): Master mode is enabled.

8.4.3.4.2 TARGET_ADDR

- **Name**: I2C Target Address Register
- **Description**: This register can be written to only when EN[0] is set to 0. You cannot change the TARGET_ADDR address dynamically.

Note:

If the software or application is aware that the I2C is not using the TARGET_ADDR address for the pending commands in the TX FIFO, it is possible to update the TARGET_ADDR address even when the TX FIFO has entries (STAT[2]= 0).

It is not necessary to perform any write to this register if the I2C is enabled as an I2C slave only.

- **Base address**: 0x4000C300 + x*0x100
- **Offset**: 0x04
- **Reset Value**: 0x00000055

Table 8-124 I2C Target Address Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R		Reserved bits
11	SPECIAL	RW	0x0	This bit indicates whether software performs a Device-ID or General Call. **Value**: • 0x0 (DISABLED): Disable programming of GENERAL_CALL or START_BYTE transmission • 0x1 (ENABLED): Enable programming of GENERAL_CALL or START_BYTE transmission
10	TX_CTRL	RW	0x0	If bit 11 (SPECIAL) is set to 1, this bit indicates whether a General Call is to be performed by the I2C.
9:0	TARGET	RW	0x55	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. If the TARGET_ADDR and S_ADDR 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; it can transmit to a slave only.

8.4.3.4.3 S_ADDR

- **Name**: I2C Slave Address Register
- **Description**: This register sets the address for I2C when it acts as a slave.
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x08
- **Reset Value**: 0x00000055

Table 8-125 I2C Slave Address Register

Bits	Field Name	RW	Reset	Description
31:10	RSVD	R		Reserved bits
9:0	S_ADDR	RW	0x55	The S_ADDR holds the slave address when the I2C is operating as a slave. For 7-bit addressing, only S_ADDR[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the EN[0] register being set to 0. Writes at other times have no effect. **Note**: The default values cannot be any of the reserved address locations: 0x00 to 0x07, or 0x78 to 0x7F. The correct operation of the device is not guaranteed if you program the S_ADDR or TARGET_ADDR to a reserved value. Refer to Table 8-121 for a complete list of these reserved values.

8.4.3.4.4 M_HS_ADDR

- **Name**: I2C High-Speed Master Mode Code Address Register
- **Description**: I2C High-Speed Master Mode Code Address Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x0C
- **Reset Value**: 0x00000001

Table 8-126 I2C High-Speed Master Mode Code Address Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	M_HS_ADDR	RW	0x1	This bit field holds the value of the I2C High-speed mode master code. High-speed-mode master codes are reserved 8-bit codes (00001xxx) that are not used for slave addressing or other purposes. Each master has its unique master code; up to eight high-speed mode masters can be present on the same I2C bus system. Valid values are from 0 to 7.

Bits	Field Name	RW	Reset	Description
				This register can be written only when the I2C interface is disabled, which corresponds to the EN[0] register being set to 0. Writes at other times have no effect.

8.4.3.4.5 DATA_CMD

- **Name**: I2C RX/TX Data Buffer and Command Register
- **Description**: This is the register the CPU writes to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO.

Note:

In order to continue acknowledging reads, a read command should be written for every byte that is to be received; otherwise the I2C will stop acknowledging.

- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x10
- **Reset Value**: 0x00000000

Table 8-127 I2C RX/TX Data Buffer and Command Register

Bits	Field Name	RW	Reset	Description
31:11	RSVD	R		Reserved bits
10	RESTART	W	0x0	This bit controls whether a RESTART is issued before the byte is sent or received. **Value**: • 0x0 (DISABLE): Do not issue RESTART before this command • 0x1 (ENABLE): Issue RESTART before this command
9	STOP	W	0x0	This bit controls whether a STOP is issued after the byte is sent or received. **Value**: • 0x0 (DISABLE): Do not issue STOP after this command • 0x1 (ENABLE): Issue STOP after this command
8	CMD	W	0x0	This bit controls whether a read or a write is performed. This bit does not control the direction when the I2C acts as a slave. It controls only the direction when it acts as a master. **Value**: • 0x0 (WRITE): Master Write command

Bits	Field Name	RW	Reset	Description
				• 0x1 (READ): Master Read command
7:0	DATA	RW	0x0	This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DATA) are ignored by the I2C. However, when you read this register, these bits return the value of data received on the I2C interface.

8.4.3.4.6 SS_CLK_HCNT

- **Name**: Standard Speed I2C Clock SCL High Count Register
- **Description**: Standard Speed I2C Clock SCL High Count Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x14
- **Reset Value**: 0x00000190

Table 8-128 Standard Speed I2C Clock SCL High Count Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	COUNT	RW	0x0190	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register can be written only when the I2C interface is disabled which corresponds to the EN[0] register 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, the value will be set to 6. **Note**: This register must not be programmed to a value higher than 65525, because I2C uses a 16-bit counter to flag an I2C bus idle condition when this counter reaches a value of COUNT + 10.

8.4.3.4.7 SS_CLK_LCNT

- **Name**: Standard Speed I2C Clock SCL Low Count Register
- **Description**: Standard Speed I2C Clock SCL Low Count Register

- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x18
- **Reset Value**: 0x000001D6

Table 8-129 Standard Speed I2C Clock SCL Low Count Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	COUNT	RW	0x01D6	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register can be written only when the I2C interface is disabled which corresponds to the EN[0] register 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, the value will be set to 8.

8.4.3.4.8 FS_CLK_HCNT

- **Name**: Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register
- **Description**: Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x1C
- **Reset Value**: 0x0000003C

Table 8-130 Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	COUNT	RW	0x003C	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register can be written only when the I2C interface is disabled, which corresponds to the EN[0] register 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, the value will be set to 6.

8.4.3.4.9 FS_CLK_LCNT

- **Name**: Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register
- **Description**: Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x20
- **Reset Value**: 0x00000082

Table 8-131 Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	COUNT	RW	0x0082	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register can be written only when the I2C interface is disabled, which corresponds to the EN[0] register 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, the value will be set to 8.

8.4.3.4.10 HS_CLK_HCNT

- **Name**: High Speed I2C Clock SCL High Count Register
- **Description**: High Speed I2C Clock SCL High Count Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x24
- **Reset Value**: 0x00000006

Table 8-132 High Speed I2C Clock SCL High Count Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	COUNT	RW	0x0006	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. The SCL High time depends on the loading of the bus. For 100 pF loading, the SCL High time is 60 ns; for 400 pF loading, the SCL High time is 120 ns.

Bits	Field Name	RW	Reset	Description
				This register can be written only when the I2C interface is disabled, which corresponds to the EN[0] register 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, the value will be set to 6.

8.4.3.4.11 HS_CLK_LCNT

- **Name**: High Speed I2C Clock SCL Low Count Register
- **Description**: High Speed I2C Clock SCL Low Count Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x28
- **Reset Value**: 0x00000010

Table 8-133 High Speed I2C Clock SCL Low Count Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	COUNT	RW	0x0010	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. The SCL low time depends on the loading of the bus. For 100pF loading, the SCL low time is 160 ns; for 400 pF loading, the SCL low time is 320 ns. This register can be written only when the I2C interface is disabled, which corresponds to the EN[0] register 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, the value will be set to 8.

8.4.3.4.12 INT_STAT

- **Name**: I2C Interrupt Status Register
- **Description**: Each bit in this register has a corresponding mask bit in the INT_MASK register. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in the RAW_INT_STAT register.

- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x2C
- **Reset Value**: 0x00000000

Table 8-134 I2C Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:15	RSVD	R		Reserved bits
14	SCL_STUCK_AT_LOW	R	0x0	See RAW_INT_STAT for a detailed description of SCL_STUCK_AT_LOW bit. **Value**: • 0x1 (ACTIVE): SCL_STUCK_AT_LOW interrupt is active. • 0x0 (INACTIVE): SCL_STUCK_AT_LOW interrupt is inactive.
13	M_HOLD	R	0x0	See RAW_INT_STAT for a detailed description of M_HOLD bit. **Value**: • 0x0 (INACTIVE): M_HOLD interrupt is inactive. • 0x1 (ACTIVE): M_HOLD interrupt is active.
12	RESTART_DET	R	0x0	See RAW_INT_STAT for a detailed description of RESTART_DET bit. **Value**: • 0x0 (INACTIVE): RESTART_DET interrupt is inactive. • 0x1 (ACTIVE): RESTART_DET interrupt is active.
11	GEN_CALL	R	0x0	See RAW_INT_STAT for a detailed description of GEN_CALL bit. **Value**: • 0x0 (INACTIVE): GEN_CALL interrupt is inactive. • 0x1 (ACTIVE): GEN_CALL interrupt is active.
10	START_DET	R	0x0	See RAW_INT_STAT for a detailed description of START_DET bit. **Value**: • 0x0 (INACTIVE): START_DET interrupt is inactive. • 0x1 (ACTIVE): START_DET interrupt is active.
9	STOP_DET	R	0x0	See RAW_INT_STAT for a detailed description of STOP_DET bit. **Value**: • 0x0 (INACTIVE): STOP_DET interrupt is inactive.

Bits	Field Name	RW	Reset	Description
				• 0x1 (ACTIVE): STOP_DET interrupt is active.
8	ACTIVITY	R	0x0	See RAW_INT_STAT for a detailed description of ACTIVITY bit. **Value**: • 0x0 (INACTIVE): ACTIVITY interrupt is inactive. • 0x1 (ACTIVE): ACTIVITY interrupt is active.
7	RX_DONE	R	0x0	See RAW_INT_STAT for a detailed description of RX_DONE bit. **Value**: • 0x0 (INACTIVE): RX_DONE interrupt is inactive. • 0x1 (ACTIVE): RX_DONE interrupt is active.
6	TX_ABORT	R	0x0	See RAW_INT_STAT for a detailed description of TX_ABORT bit. **Value**: • 0x0 (INACTIVE): TX_ABORT interrupt is inactive. • 0x1 (ACTIVE): TX_ABORT interrupt is active.
5	RD_REQ	R	0x0	See RAW_INT_STAT for a detailed description of RD_REQ bit. **Value**: • 0x0 (INACTIVE): RD_REQ interrupt is inactive. • 0x1 (ACTIVE): RD_REQ interrupt is active.
4	TX_EMPTY	R	0x0	See RAW_INT_STAT for a detailed description of TX_EMPTY bit. **Value**: • 0x0 (INACTIVE): TX_EMPTY interrupt is inactive. • 0x1 (ACTIVE): TX_EMPTY interrupt is active.
3	TX_OVER	R	0x0	See RAW_INT_STAT for a detailed description of TX_OVER bit. **Value**: • 0x0 (INACTIVE): TX_OVER interrupt is inactive. • 0x1 (ACTIVE): TX_OVER interrupt is active.
2	RX_FULL	R	0x0	See RAW_INT_STAT for a detailed description of RX_FULL bit. **Value**: • 0x0 (INACTIVE): RX_FULL interrupt is inactive.

Bits	Field Name	RW	Reset	Description
				• 0x1 (ACTIVE): RX_FULL interrupt is active.
1	RX_OVER	R	0x0	See RAW_INT_STAT for a detailed description of RX_OVER bit. **Value**: • 0x0 (INACTIVE): RX_OVER interrupt is inactive. • 0x1 (ACTIVE): RX_OVER interrupt is active.
0	RX_UNDER	R	0x0	See RAW_INT_STAT for a detailed description of RX_UNDER bit. **Value**: • 0x0 (INACTIVE): RX_UNDER interrupt is inactive. • 0x1 (ACTIVE): RX_UNDER interrupt is active.

8.4.3.4.13 INT_MASK

- **Name**: I2C Interrupt Mask Register
- **Description**: 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.
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x30
- **Reset Value**: 0x000008FF

Table 8-135 I2C Interrupt Mask Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R		Reserved bits
11	MASK_GEN_CALL	RW	0x1	This bit masks the GEN_CALL interrupt in INT_STAT register. **Value**: • 0x0 (ENABLED): GEN_CALL interrupt is masked. • 0x1 (DISABLED): GEN_CALL interrupt is unmasked.
10	MASK_START_DET	RW	0x0	This bit masks the START_DET interrupt in INT_STAT register. **Value**: • 0x0 (ENABLED): START_DET interrupt is masked. • 0x1 (DISABLED): START_DET interrupt is unmasked.
9	MASK_STOP_DET	RW	0x0	This bit masks the STOP_DET interrupt in INT_STAT register. **Value**: • 0x0 (ENABLED): STOP_DET interrupt is masked.

Bits	Field Name	RW	Reset	Description
				• 0x1 (DISABLED): STOP_DET interrupt is unmasked.
8	MASK_ACTIVITY	RW	0x0	This bit masks the ACTIVITY interrupt in INT_STAT register. **Value**: • 0x0 (ENABLED): ACTIVITY interrupt is masked. • 0x1 (DISABLED): ACTIVITY interrupt is unmasked.
7	MASK_RX_DONE	RW	0x1	This bit masks the RX_DONE interrupt in INT_STAT register. **Value**: • 0x0 (ENABLED): RX_DONE interrupt is masked. • 0x1 (DISABLED): RX_DONE interrupt is unmasked.
6	MASK_TX_ABORT	RW	0x1	This bit masks the TX_ABORT interrupt in INT_STAT register. **Value**: • 0x0 (ENABLED): TX_ABORT interrupt is masked. • 0x1 (DISABLED): TX_ABORT interrupt is unmasked.
5	MASK_RD_REQ	RW	0x1	This bit masks the RD_REQ interrupt in INT_STAT register. **Value**: • 0x0 (ENABLED): RD_REQ interrupt is masked. • 0x1 (DISABLED): RD_REQ interrupt is unmasked.
4	MASK_TX_EMPTY	RW	0x1	This bit masks the TX_EMPTY interrupt in INT_STAT register. **Value**: • 0x0 (ENABLED): TX_EMPTY interrupt is masked. • 0x1 (DISABLED): TX_EMPTY interrupt is unmasked.
3	MASK_TX_OVER	RW	0x1	This bit masks the TX_OVER interrupt in INT_STAT register. **Value**: • 0x0 (ENABLED): TX_OVER interrupt is masked. • 0x1 (DISABLED): TX_OVER interrupt is unmasked.
2	MASK_RX_FULL	RW	0x1	This bit masks the RX_FULL interrupt in INT_STAT register. **Value**: • 0x0 (ENABLED): RX_FULL interrupt is masked. • 0x1 (DISABLED): RX_FULL interrupt is unmasked.
1	MASK_RX_OVER	RW	0x1	This bit masks the RX_OVER interrupt in INT_STAT register. **Value**: • 0x0 (ENABLED): RX_OVER interrupt is masked. • 0x1 (DISABLED): RX_OVER interrupt is unmasked.
0	MASK_RX_UNDER	RW	0x1	This bit masks the RX_UNDER interrupt in INT_STAT register.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (ENABLED): RX_UNDER interrupt is masked. • 0x1 (DISABLED): RX_UNDER interrupt is unmasked.

8.4.3.4.14 RAW_INT_STAT

- **Name**: I2C Raw Interrupt Status Register
- **Description**: Unlike the INT_STAT register, these bits are not masked so they always show the true status of the I2C.
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x34
- **Reset Value**: 0x00000000

Table 8-136 I2C Raw Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:15	RSVD	R		Reserved bits
14	RAW_SCL_STUCK_AT_LOW	R	0x0	Indicates whether the SCL Line is stuck at low for the SCL_STUCK_LOW_TIMEOUT number of ic_clk periods **Value**: • 0x0 (ACTIVE): RAW_SCL_STUCK_AT_LOW interrupt is active. • 0x1 (ACTIVE): RAW_SCL_STUCK_AT_LOW interrupt is inactive.
13	RAW_M_HOLD	R	0x0	Indicate whether the master is holding the bus and TX FIFO is empty. **Value**: • 0x1 (ACTIVE): RAW_M_HOLD interrupt is active • 0x0 (INACTIVE): RAW_M_HOLD interrupt is inactive
12	RAW_RESTART_DET	R	0x0	Indicate whether a RESTART condition has occurred on the I2C interface when I2C is operating in Slave mode and the slave is being addressed. **Note:** However, in High-speed Mode, the RESTART comes before the address field as per the I2C protocol. In this case, the slave is not the addressed slave when the RESTART is issued, therefore I2C does not generate the RESTART_DET interrupt.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (INACTIVE): RAW_RESTART_DET interrupt is inactive • 0x1 (ACTIVE): RAW_RESTART_DET interrupt is active
11	RAW_GEN_CALL	R	0x0	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling I2C or when the CPU reads bit 0 of the CLR_GEN_CALL register. I2C stores the received data in the RX buffer. **Value**: • 0x0 (INACTIVE): RAW_GEN_CALL interrupt is inactive • 0x1 (ACTIVE): RAW_GEN_CALL interrupt is active
10	RAW_START_DET	R	0x0	Indicate whether a START or RESTART condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode. **Value**: • 0x0 (INACTIVE): RAW_START_DET interrupt is inactive • 0x1 (ACTIVE): RAW_START_DET interrupt is active
9	RAW_STOP_DET	R	0x0	Indicate whether a STOP condition has occurred on the I2C interface regardless of whether I2C is operating in slave or master mode. In Slave Mode: If CTRL[7] = 0x1 (STOP_DET_INT), the STOP_DET interrupt will be issued only if slave is addressed. **Note**: During a general call address, this slave does not issue a STOP_DET interrupt if STOP_DET_INT = 0x1, even if the slave responds to the general call address by generating ACK. The STOP_DET interrupt is generated only when the transmitted address matches the slave address (S_ADDR). If CTRL[7] = 0x0 (STOP_DET_INT), the STOP_DET interrupt is issued irrespective of whether it is being addressed. **In Master Mode**: • If CTRL[10] = 0x1, the STOP_DET interrupt will be issued only if Master is active.

Bits	Field Name	RW	Reset	Description
				• If CTRL[10] = 0x0, the STOP_DET interrupt will be issued irrespective of whether master is active or not. **Value**: • 0x0 (INACTIVE): RAW_STOP_DET interrupt is inactive. • 0x1 (ACTIVE): RAW_STOP_DET interrupt is active.
8	RAW_ACTIVITY	R	0x0	This bit captures I2C activity and stays set until it is cleared. There are four ways to clear it: • Disabling the I2C • Reading the CLR_ACTIVITY register • Reading the CLR_INT register • System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the I2C module is idle, this bit remains set until cleared, indicating that there was activity on the bus. **Value**: • 0x0 (INACTIVE): RAW_ACTIVITY interrupt is inactive. • 0x1 (ACTIVE): RAW_ACTIVITY interrupt is active.
7	RAW_RX_DONE	R	0x0	When the I2C 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. **Value**: • 0x0 (INACTIVE): RAW_RX_DONE interrupt is inactive. • 0x1 (ACTIVE): RAW_RX_DONE interrupt is active.
6	RAW_TX_ABORT	R	0x0	This bit indicates if I2C, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a 'transmit abort'. When this bit is set to 1, the TX_ABORT_SRC register indicates the reason why the transmit abort takes place. **Note**: The I2C flushes/resets/empties only the TX_FIFO whenever there is a transmit abort caused by any of the events tracked

Bits	Field Name	RW	Reset	Description
				by the TX_ABORT_SRC register. The TX FIFO remains in this flushed state until the register CLR_TX_ABORT is read. Once this read is performed, the TX FIFO is then ready to accept more data bytes from the APB interface. **Value**: • 0x0 (INACTIVE): RAW_TX_ABORT interrupt is inactive. • 0x1 (ACTIVE): RAW_TX_ABORT interrupt is active.
5	RAW_RD_REQ	R	0x0	This bit is set to 1 when I2C is acting as a slave and another I2C master is attempting to read data from I2C. The I2C holds the I2C 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 DATA_CMD register. This bit is set to 0 just after the processor reads the CLR_RD_REQ register. **Value**: • 0x0 (INACTIVE): RAW_RD_REQ interrupt is inactive. • 0x1 (ACTIVE): RAW_RD_REQ interrupt is active.
4	RAW_TX_EMPTY	R	0x0	The behavior of the TX_EMPTY interrupt status differs based on the TX_EMPTY_CTRL selection in the CTRL register. • When TX_EMPTY_CTRL = 0: This bit is set to 1 when the transmit buffer is at or below the threshold value set in the TX_FIFO_THD register. • When TX_EMPTY_CTRL = 1: This bit is set to 1 when the transmit buffer is at or below the threshold value set in the TX_FIFO_THD register and the transmission of the address/data from the internal shift register for the most recently popped command is completed. It is automatically cleared by hardware when the buffer level goes above the threshold. When EN[0] is set to 0, the TX FIFO is flushed and held in reset. Therefore, the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is any activity in the master or slave state machines. When there is no longer any activity, then with EN[0] = 0, this bit is set to 0.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (INACTIVE): RAW_TX_EMPTY interrupt is inactive. • 0x1 (ACTIVE): RAW_TX_EMPTY interrupt is active.
3	RAW_TX_OVER	R	0x0	Set during transmit if the transmit buffer is filled to 8 and the processor attempts to issue another I2C command by writing to the DATA_CMD register. When I2C module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when EN[0] goes to 0, this interrupt is cleared. **Value**: • 0x0 (INACTIVE): RAW_TX_OVER interrupt is inactive • 0x1 (ACTIVE): RAW_TX_OVER interrupt is active
2	RAW_RX_FULL	R	0x0	Set when the receive buffer reaches or goes above the RX_FIFO_THD threshold in the RX_FIFO_THD register. It is automatically cleared by hardware when buffer level goes below the threshold. If I2C module is disabled (EN[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 EN bit 0 is programmed with a 0, regardless of the activity that continues. **Value**: • 0x0 (INACTIVE): RAW_RX_FULL interrupt is inactive. • 0x1 (ACTIVE): RAW_RX_FULL interrupt is active.
1	RAW_RX_OVER	R	0x0	Set if the receive buffer is completely filled to 8 and an additional byte is received from an external I2C device. The I2C acknowledges this, but any data bytes received after the FIFO is full are lost. If I2C module is disabled (EN[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when EN[0] goes to 0, this interrupt is cleared. **Value**: • 0x0 (INACTIVE): RAW_RX_OVER interrupt is inactive. • 0x1 (ACTIVE): RAW_RX_OVER interrupt is active.
0	RAW_RX_UNDER	R	0x0	Set if the processor attempts to read the receive buffer when it is empty by reading from the DATA_CMD register. If the module is disabled (EN[0]=0), this bit keeps its level until

Bits	Field Name	RW	Reset	Description
				the master or slave state machines go into idle, and when EN[0] goes to 0, this interrupt is cleared. **Value**: • 0x0 (INACTIVE): RAW_RX_UNDER interrupt is inactive. • 0x1 (ACTIVE): RAW_RX_UNDER interrupt is active.

8.4.3.4.15 RX_FIFO_THD

- **Name**: I2C Receive FIFO Threshold Register
- **Description**: I2C Receive FIFO Threshold Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x38
- **Reset Value**: 0x00000000

Table 8-137 I2C Receive FIFO Threshold Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	THD	RW	0x00	Receive FIFO Threshold Level. Control the level of entries (or above) that triggers the RAW_RX_FULL interrupt (bit 2 in RAW_INT_STAT register). The valid range is 0–255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries.

8.4.3.4.16 TX_FIFO_THD

- **Name**: I2C Transmit FIFO Threshold Register
- **Description**: I2C Transmit FIFO Threshold Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x3C
- **Reset Value**: 0x00000000

Table 8-138 I2C Transmit FIFO Threshold Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	THD	RW	0x00	Transmit FIFO Threshold Level. Control the level of entries (or below) that trigger the RAW_TX_EMPTY interrupt (bit 4 in RAW_INT_STAT register). The valid range is 0–255, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 255 sets the threshold for 255 entries.

8.4.3.4.17 CLR_INT

- **Name**: Clear Combined and Individual Interrupt Register
- **Description**: Clear Combined and Individual Interrupt Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x40
- **Reset Value**: 0x00000000

Table 8-139 Clear Combined and Individual Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_INT	R	0x0	Read this register to clear the combined interrupt, all individual interrupts, and the TX_ABORT_SRC register. This bit does not clear hardware clearable interrupts but software clearable interrupts.

8.4.3.4.18 CLR_RX_UNDER

- **Name**: Clear RX_UNDER Interrupt Register
- **Description**: Clear RX_UNDER Interrupt Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x44
- **Reset Value**: 0x00000000

Table 8-140 Clear RX_UNDER Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_RX_UNDER	R	0x0	Read this register to clear the RAW_RX_UNDER interrupt (bit 0) of the RAW_INT_STAT register.

8.4.3.4.19 CLR_RX_OVER

- **Name**: Clear RX_OVER Interrupt Register
- **Description**: Clear RX_OVER Interrupt Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x48
- **Reset Value**: 0x00000000

Table 8-141 Clear RX_OVER Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_RX_OVER	R	0x0	Read this register to clear the RAW_RX_OVER interrupt (bit 1) of the RAW_INT_STAT register.

8.4.3.4.20 CLR_TX_OVER

- **Name**: Clear TX_OVER Interrupt Register
- **Description**: Clear TX_OVER Interrupt Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x4C
- **Reset Value**: 0x00000000

Table 8-142 Clear TX_OVER Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_TX_OVER	R	0x0	Read this register to clear the RAW_TX_OVER interrupt (bit 3) of the RAW_INT_STAT register.

8.4.3.4.21 CLR_RD_REQ

- **Name**: Clear RD_REQ Interrupt Register

- **Description**: Clear RD_REQ Interrupt Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x50
- **Reset Value**: 0x00000000

Table 8-143 Clear RD_REQ Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_RD_REQ	R	0x0	Read this register to clear the RAW_RD_REQ interrupt (bit 5) of the RAW_INT_STAT register.

8.4.3.4.22 CLR_TX_ABORT

- **Name**: Clear TX_ABORT Interrupt Register
- **Description**: Clear TX_ABORT Interrupt Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x54
- **Reset Value**: 0x00000000

Table 8-144 Clear TX_ABORT Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_TX_ABORT	R	0x0	Read this register to clear the RAW_TX_ABORT interrupt (bit 6) of the RAW_INT_STAT register, and the TX_ABORT_SRC register.

8.4.3.4.23 CLR_RX_DONE

- **Name**: Clear RX_DONE Interrupt Register
- **Description**: Clear RX_DONE Interrupt Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x58
- **Reset Value**: 0x00000000

Table 8-145 Clear RX_DONE Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
0	CLR_RX_DONE	R	0x0	Read this register to clear the RAW_RX_DONE interrupt (bit 7) of the RAW_INT_STAT register.

8.4.3.4.24 CLR_ACTIVITY

- **Name**: Clear ACTIVITY Interrupt Register
- **Description**: Clear ACTIVITY Interrupt Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x5C
- **Reset Value**: 0x00000000

Table 8-146 Clear ACTIVITY Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_ACTIVITY	R	0x0	Reading this register clears the ACTIVITY interrupt if the I2C is not active anymore. If the I2C 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 RAW_ACTIVITY interrupt (bit 8) of the RAW_INT_STAT register.

8.4.3.4.25 CLR_STOP_DET

- **Name**: Clear STOP_DET Interrupt Register
- **Description**: Clear STOP_DET Interrupt Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x60
- **Reset Value**: 0x00000000

Table 8-147 Clear STOP_DET Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_STOP_DET	R	0x0	Read this register to clear the RAW_STOP_DET interrupt (bit 9) of the RAW_INT_STAT register.

8.4.3.4.26 CLR_START_DET

- **Name**: Clear START_DET Interrupt Register
- **Description**: Clear START_DET Interrupt Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x64
- **Reset Value**: 0x00000000

Table 8-148 Clear START_DET Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_START_DET	R	0x0	Read this register to clear the RAW_START_DET interrupt (bit 10) of the RAW_INT_STAT register.

8.4.3.4.27 CLR_GEN_CALL

- **Name**: Clear GEN_CALL Interrupt Register
- **Description**: Clear GEN_CALL Interrupt Register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x68
- **Reset Value**: 0x00000000

Table 8-149 Clear GEN_CALL Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_GEN_CALL	R	0x0	Read this register to clear the RAW_GEN_CALL interrupt (bit 11) of RAW_INT_STAT register.

8.4.3.4.28 EN

- **Name**: I2C ENABLE Register
- **Description**: I2C enable register
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x6C
- **Reset Value**: 0x00000000

Table 8-150 I2C ENABLE Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3	SDA_STUCK_RECOVERY_ENABLE	RW	0x0	If SDA is stuck at low, which is indicated by the TX_ABORT interrupt (TX_ABORT_SRC[17]), this bit is used as a control knob to initiate the SDA Recovery Mechanism (that is, send at most 9 SCL clocks and STOP to release the SDA line) and then this bit gets auto clear. **Value**: • 0x1 (SDA_STUCK_RECOVERY_ENABLED): Master initiates the SDA stuck at low recovery mechanism. • 0x0 (SDA_STUCK_RECOVERY_DISABLED): Master disables the SDA stuck at low recovery mechanism.
2	TX_CMD_BLOCK	RW	0x0	In Master mode: **Value**: • 0x0 (NOT_BLOCKED): TX Command execution not blocked • 0x1 (BLOCKED): TX Command execution blocked
1	ABORT	RW	0x0	When set, the controller initiates the transfer abort. The software can abort the I2C transfer in master mode by setting this bit. The software can set this bit only when EN 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 current transfer, then sets the TX_ABORT interrupt after the abort operation. The ABORT bit is cleared automatically after the abort operation. **Value**: • 0x0 (DISABLED): ABORT operation not in progress • 0x1 (ENABLED): ABORT operation in progress
0	ACTIVITY	RW	0x0	Control whether the I2C is enabled. Software can disable I2C while it is active. However, it is important that care should be taken to ensure that I2C is disabled properly. When I2C is disabled, the following occurs: • The TX FIFO and RX FIFO get flushed.

Bits	Field Name	RW	Reset	Description
				• Status bits in the INT_STAT register are still active until I2C goes into IDLE state. If the I2C module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is completed. If the module is receiving, the I2C stops the current transfer at the end of the current byte and does not acknowledge the transfer. **Value**: • 0x0 (DISABLED): I2C is disabled. • 0x1 (ENABLED): I2C is enabled.

8.4.3.4.29 STAT

- **Name**: I2C STATUS Register
- **Description**: 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 requests an interrupt.

  When the I2C is disabled by writing 0 in Bit 0 of the EN register:
  - Bits 1 and 2 are set to 1
  - Bits 3 and 6 are set to 0
  - When the master or slave state machine goes idle and EN[0]=0:
  - Bits 5 and 6 are set to 0
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x70
- **Reset Value**: 0x00000006

Table 8-151 I2C STATUS Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R		Reserved bits
11	SDA_STUCK_NOT_RECOVERED	R	0x0	This bit indicates that SDA stuck at low is not recovered after the recovery mechanism. In Slave mode, this register bit is not applicable. **Value**: • 0x1 (ACTIVE): SDA Stuck at low is recovered after recovery mechanism.

Bits	Field Name	RW	Reset	Description
				• 0x0 (INACTIVE): SDA Stuck at low is not recovered after recovery mechanism.
10:7	RSVD	R		Reserved bits
6	S_ACTIVITY	R	0x0	Slave Activity Status. When the Slave is not in the IDLE state, this bit is set. **Value**: • 0x0 (INACTIVE): Slave is idle. • 0x1 (ACTIVE): Slave is not idle.
5	M_ACTIVITY	R	0x0	Master Activity Status. When the Master is not in the IDLE state, this bit is set. **Note**: STAT[0], ACTIVITY bit, is the OR of S_ACTIVITY and M_ACTIVITY Y bits. **Value**: • 0x0 (INACTIVE): Master is idle. • 0x1 (ACTIVE): Master not idle.
4	RX_FIFO_CF	R	0x0	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. **Value**: • 0x0 (NOT_FULL): RX FIFO is not full. • 0x1 (FULL): RX FIFO is full.
3	RX_FIFO_NE	R	0x0	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. **Value**: • 0x0 (EMPTY): RX FIFO is empty. • 0x1 (NOT_EMPTY): RX FIFO not empty.
2	TX_FIFO_CE	R	0x1	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. **Value**: • 0x0 (NOT_EMPTY): TX FIFO not empty. • 0x1 (EMPTY): TX FIFO is empty.

Bits	Field Name	RW	Reset	Description
1	TX_FIFO_NF	R	0x1	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. **Value**: • 0x0 (FULL): TX FIFO is full. • 0x1 (NOT_FULL): TX FIFO not full.
0	ACTIVITY	R	0x0	I2C Activity Status **Value**: • 0x0 (INACTIVE): I2C is idle. • 0x1 (ACTIVE): I2C is active.

8.4.3.4.30 TX_FIFO_LV

- **Name**: I2C Transmit FIFO Level Register
- **Description**: This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:
  - The I2C is disabled.
  - If RAW_TX_ABORT bit is set in the RAW_INT_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.
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x74
- **Reset Value**: 0x00000000

Table 8-152 I2C Transmit FIFO Level Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	TLEVEL	R	0x0	Transmit FIFO Level. It contains the number of valid data entries in the transmit FIFO.

8.4.3.4.31 RX_FIFO_LV

- **Name**: I2C Receive FIFO Level Register
- **Description**: This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:
  - The I2C is disabled.

- There is a transmit abort caused by any of the events tracked in TX_ABORT_SRC.
- The register increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.

- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x78
- **Reset Value**: 0x00000000

Table 8-153 I2C Receive FIFO Level Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	LEVEL	R	0x0	Receive FIFO Level. It contains the number of valid data entries in the receive FIFO.

8.4.3.4.32 SDA_HOLD

- **Name**: I2C SDA Hold Time Length Register
- **Description**: I2C SDA Hold Time Length Register
  - The bits [15:0] of this register are used to control the hold time of SDA during transmit in both slave and master mode (after SCL goes from HIGH to LOW).
  - The bits [23:16] of this register are used to extend the SDA transition (if any) whenever SCL is HIGH in the receiver in either master or slave mode.
  - Writes to this register succeed only when EN[0]=0.
  - The values in this register are in units of clock period. The value programmed in TX_HOLD must be greater than the minimum hold time in each mode one cycle in master mode, seven cycles in slave mode for the value to be implemented.
  - The programmed SDA hold time during transmit (TX_HOLD) cannot exceed at any time the duration of the low part of SCL. Therefore the programmed value cannot be larger than N_SCL_LOW minus 2, where N_SCL_LOW is the duration of the low part of the SCL period measured in clock cycles.
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x7C
- **Reset Value**: 0x00000001

Table 8-154 I2C SDA Hold Time Length Register

Bits	Field Name	RW	Reset	Description
31:24	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
23:16	RX_HOLD	RW	0x0	Set the required SDA hold time in units of clock period, when I2C acts as a receiver.
15:0	TX_HOLD	RW	0x1	Set the required SDA hold time in units of clock period, when I2C acts as a transmitter.

8.4.3.4.33 TX_ABORT_SRC

- **Name**: I2C Transmit Abort Source Register
- **Description**: I2C Transmit Abort Source Register
  - This register has 32 bits that indicate the source of the TX_ABORT bit. Except for Bit 9, this register is cleared whenever the CLR_TX_ABORT register bit or the CLR_INT register is read. To clear Bit 9, the source of ABORT_SBYTE_NORSTRT must be fixed first; RESTART must be enabled (CTRL[5] = 1), the SPECIAL bit must be cleared (TARGET_ADDR[11]), or the TX_CTRL bit must be cleared (TARGET_ADDR[10]).
  - Once the source of the ABORT_SBYTE_NORSTRT is fixed, this bit can be cleared in the same manner as other bits in this register. If the source of the ABORT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x80
- **Reset Value**: 0x00000000

Table 8-155 I2C Transmit Abort Source Register

Bits	Field Name	RW	Reset	Description
31:23	TX_FLUSH_CNT	R	0x0	This field indicates the number of TX FIFO Data Commands which are flushed due to TX_ABORT interrupt. It is cleared whenever I2C is disabled. Role of I2C: Master-Transmitter or Slave-Transmitter
22:18	RSVD	R		Reserved bits
17	ABRT_SDA_STUCK_AT_LOW	R	0x0	This is a master-mode-only bit. Master detects the SDA Stuck at low for the SDA_STUCK_AT_LOW_TIMEOUT value of ic_clk cycles. **Value**: • 0x1 (ACTIVE): This abort is generated because of SDA stuck at low for SDA_STUCK_AT_LOW_TIMEOUT value of ic_clk cycles. • 0x0 (INACTIVE): This abort is not generated.

Bits	Field Name	RW	Reset	Description
16	ABORT_USER_ABORT	R	0x0	This is a master-mode-only bit. Master has detected the transfer abort (EN[1]) event. Role of I2C: Master-Transmitter **Value**: • 0x0 (ABORT_USER_ABORT_VOID): Transfer abort not detected by master • 0x1 (ABORT_USER_ABORT_GENERATED): Transfer abort detected by master
15	ABORT_SLVRD_INTX	R	0x0	When the processor side responds to a slave mode request for data to be transmitted to a remote master and users write a 1 in CMD (bit 8) of DATA_CMD register. Role of I2C: Slave-Transmitter **Value**: • 0x0 (ABORT_SLVRD_INTX_VOID): Slave trying to transmit to remote master in read mode- scenario not present • 0x1 (ABORT_SLVRD_INTX_GENERATED): Slave trying to transmit to remote master in read mode
14	ABORT_S_ARBLOST	R	0x0	This field indicates that a Slave has lost the bus while transmitting data to a remote master. TX_ABORT_SRC[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 I2C no longer owns the bus. Role of I2C: Slave-Transmitter **Value**: • 0x0 (ABORT_S_ARBLOST_VOID): Slave lost arbitration to remote master- scenario not present • 0x1 (ABORT_S_ARBLOST_GENERATED): Slave lost arbitration to remote master
13	ABORT_SLVFLUSH_TXFIFO	R	0x0	This field specifies that the Slave has received a read command and some data exists in the TX FIFO, so the slave issues a TX_ABORT interrupt to flush old data in TX FIFO.

Bits	Field Name	RW	Reset	Description
				Role of I2C: Slave-Transmitter **Value**: • 0x0 (ABORT_SLVFLUSH_TXFIFO_VOID): Slave flushes existing data in TX-FIFO upon getting read command- scenario not present • 0x1 (ABORT_SLVFLUSH_TXFIFO_GENERATED): Slave flushes existing data in TX-FIFO upon getting read command
12	ABORT_LOST	R	0x0	This field specifies that the Master has lost arbitration, or if TX_ABORT_SRC[14] is also set, then the slave transmitter has lost arbitration. Role of I2C: Master-Transmitter or Slave-Transmitter **Value**: • 0x0 (ABORT_LOST_VOID): Master or Slave-Transmitter lost arbitration- scenario not present • 0x1 (ABORT_LOST_GENERATED): Master or Slave-Transmitter lost arbitration
11	ABORT_MASTER_DIS	R	0x0	This field indicates that users try to initiate a Master operation with the Master mode disabled. Role of I2C: Master-Transmitter or Master-Receiver **Value**: • 0x0 (ABORT_MASTER_DIS_VOID): Users initiates master operation when Master is disabled- scenario not present • 0x1 (ABORT_MASTER_DIS_GENERATED): Users initiates master operation when Master is disabled
10	ABORT_10B_RD_NORSTR	R	0x0	This field indicates that the restart is disabled (RESTART_EN bit (CTRL[5]) =0) and the master sends a read command in 10-bit addressing mode. Role of I2C: Master-Receiver **Value**: • 0x0 (ABRT_10B_RD_VOID): Master not trying to read in 10-bit addressing mode when RESTART disabled • 0x1 (ABRT_10B_RD_GENERATED): Master trying to read in 10-bit addressing mode when RESTART disabled
9	ABORT_SBYTE_NORSTRT	R	0x0	To clear Bit 9, the source of the ABORT_SBYTE_NORSTRT must be fixed first; RESTART must be enabled (RESTART_EN bit CTRL[5]=1), the SPECIAL bit must be cleared

Bits	Field Name	RW	Reset	Description
				(TARGET_ADDR[11]), or the TX_CTRL bit must be cleared (TARGET_ADDR[10]). Once the source of the ABORT_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 ABORT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets reasserted. When this field is set to 1, RESTART is disabled (RESTART_EN bit (CTRL[5]) =0) and users are trying to send a START BYTE. Role of I2C: Master **Value**: • 0x0 (ABORT_SBYTE_NORSTRT_VOID): Users trying to send START BYTE when RESTART is disabled- scenario not present • 0x1 (ABORT_SBYTE_NORSTRT_GENERATED): Users trying to send START BYTE when RESTART is disabled
8	ABORT_HS_NORSTRT	R	0x0	This field indicates that the RESTART is disabled (RESTART_EN bit (CTRL[5]) =0) and users are trying to use the master to transfer data in High-speed mode. Role of I2C: Master-Transmitter or Master-Receiver **Value**: • 0x0 (ABORT_HS_NORSTRT_VOID): Users trying to switch Master to High-speed mode when RESTART is disabled- scenario not present • 0x1 (ABORT_HS_NORSTRT_GENERATED): Users trying to switch Master to High-speed mode when RESTART is disabled
7	ABORT_SBYTE_ACKDET	R	0x0	This field indicates that the Master has sent a START Byte and the START Byte was acknowledged (wrong behavior). Role of I2C: Master **Value**: • 0x0 (ABORT_SBYTE_ACKDET_VOID): ACK detected for START byte- scenario not present • 0x1 (ABORT_SBYTE_ACKDET_GENERATED): ACK detected for START byte

Bits	Field Name	RW	Reset	Description
6	ABORT_HS_ACKDET	R	0x0	This field indicates that the Master is in High-speed mode and the High-speed Master code was acknowledged (wrong behavior). Role of I2C: Master **Value**: • 0x0 (ABRT_HS_ACK_VOID): HS Master code ACKed in High-speed Mode- scenario not present • 0x1 (ABRT_HS_ACK_GENERATED): High-speed Master code ACKed in High-speed Mode
5	ABORT_GCALL_RD	R	0x0	This field indicates that I2C in the master mode has sent a General Call but users program the byte that follows the General Call to be a read from the bus (DATA_CMD[9] is set to 1). Role of I2C: Master-Transmitter **Value**: • 0x0 (ABORT_GCALL_RD_VOID): GCALL is followed by read from bus-scenario not present • 0x1 (ABORT_GCALL_RD_GENERATED): GCALL is followed by read from bus
4	ABORT_GCALL_NOACK	R	0x0	This field indicates that I2C in master mode has sent a General Call and no slave on the bus acknowledged the General Call. Role of I2C: Master-Transmitter **Value**: • 0x0 (ABORT_GCALL_NOACK_VOID): GCALL not ACKed by any slave-scenario not present • 0x1 (ABORT_GCALL_NOACK_GENERATED): GCALL not ACKed by any slave
3	ABORT_TX_NOACK	R	0x0	This field indicates the master-mode only bit. When the master receives an acknowledgement for the address, but when it sends data byte(s) following the address, it did not receive an acknowledge from the remote slave(s). Role of I2C: Master-Transmitter **Value**: • 0x0 (ABORT_TX_NOACK_VOID): Transmitted data non-ACKed by addressed slave-scenario not present

Bits	Field Name	RW	Reset	Description
				• 0x1 (ABORT_TX_NOACK_GENERATED): Transmitted data not ACKed by addressed slave
2	ABORT_10B2_NOACK	R	0x0	This field indicates that the Master is in 10-bit address mode and that the second address byte of the 10-bit address was not acknowledged by any slave. Role of I2C: Master-Transmitter or Master-Receiver **Value**: • 0x0 (INACTIVE): This abort is not generated. • 0x1 (ACTIVE): Byte 2 of 10-bit Address is not ACKed by any slave.
1	ABORT_10B1_NOACK	R	0x0	This field indicates that the Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave. Role of I2C: Master-Transmitter or Master-Receiver **Value**: • 0x0 (INACTIVE): This abort is not generated. • 0x1 (ACTIVE): Byte 1 of 10-bit Address is not ACKed by any slave.
0	ABORT_7B_NOACK	R	0x0	This field indicates that the Master is in 7-bit address mode and the address sent was not acknowledged by any slave. Role of I2C: Master-Transmitter or Master-Receiver **Value**: • 0x0 (INACTIVE): This abort is not generated. • 0x1 (ACTIVE): This abort is generated because of NACK for 7-bit address.

8.4.3.4.34 DMA_CTRL

- **Name**: DMA Control Register
- **Description**: This register is only valid when I2C is configured with a set of DMA Controller interface signals. When I2C is not configured for DMA operation, this register does not exist and writing to the register address has no effect and reading from this register address will return zero. The register is used to enable the DMA Controller interface operation. There is a separate bit for transmitting and receiving. This can be programmed regardless of the state of EN.
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x88
- **Reset Value**: 0x00000000

Table 8-156 DMA Control Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	TX_EN	RW	0x0	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. **Value**: • 0x0 (DISABLED): Transmit FIFO DMA channel disabled • 0x1 (ENABLED): Transmit FIFO DMA channel enabled
0	RX_EN	RW	0x0	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. **Value**: • 0x0 (DISABLED): Receive FIFO DMA channel disabled • 0x1 (ENABLED): Receive FIFO DMA channel enabled

8.4.3.4.35 DMA_TX_LV

- **Name**: DMA Transmit Data Level Register
- **Description**: This register is only valid when the I2C is configured with a set of DMA interface signals. When I2C is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x8C
- **Reset Value**: 0x00000000

Table 8-157 DMA Transmit Data Level Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	LEVEL	RW	0x0	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic.

8.4.3.4.36 DMA_RX_LV

- **Name**: DMA Receive Data Level Register
- **Description**: This register is only valid when I2C is configured with a set of DMA interface signals. When I2C is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.
- **Base Address**: 0x4000C300 + x*0x100

- **Offset**: 0x90
- **Reset Value**: 0x00000000

Table 8-158 DMA Receive Data Level Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	LEVEL	RW	0x0	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic.

8.4.3.4.37 SDA_SETUP

- **Name**: I2C SDA Setup Register
- **Description**: This register controls the amount of time delay (in terms of number of clock periods) introduced in the rising edge of SCL – relative to SDA changing – when I2C services a read request in a slave-transmitter operation. The relevant I2C requirement is tSU_STO as detailed in the Figure 8-35. This register must be programmed with a value that is equal to or greater than 2.

  Writes to this register succeed only when EN[0] = 0.

Note:

The length of setup time is calculated using [(SDA_SETUP – 1)*(clk_period)], so if users require 10 clock periods of setup time, they should program a value of 11. The SDA_SETUP register is only used by the I2C when operating as a slave transmitter.

- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x94
- **Reset Value**: 0x00000064

Table 8-159 I2C SDA Setup Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	SETUP	RW	0x64	SDA Setup. It is recommended that if the required delay is 1000 ns, then for a clock frequency of 10 MHz, SETUP should be programmed to a value of 11. SETUP must be programmed with a minimum value of 2.

8.4.3.4.38 ACK_GEN_CALL

- **Name**: I2C ACK General Call Register

- **Description**: The register controls whether I2C responds with an ACK or NACK when it receives an I2C General Call address. This register is applicable only when the I2C is in slave mode.
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x98
- **Reset Value**: 0x00000001

Table 8-160 I2C ACK General Call Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	ACK_GEN_CALL	RW	0x1	ACK General Call. When this bit is set to 1, I2C responds with an ACK (by asserting data_oe) when it receives a General Call. Otherwise, I2C responds with a NACK (by negating data_oe). **Value**: • 0x0 (DISABLED): Generate NACK for General Call. • 0x1 (ENABLED): Generate ACK for a General Call.

8.4.3.4.39 EN_STAT

- **Name**: I2C Enable Status Register
- **Description**: The register is used to report the I2C hardware status when the EN[0] register bit is set from 1 to 0 (when I2C is disabled).
  - If EN[0] has been set to 1, Bits [2:1] of the EN_STAT register are forced to 0, and Bit 0 is forced to 1.
  - If EN[0] has been set to 0, Bits [2:1] of the EN_STAT register are only valid as soon as Bit 0 is read as '0'.

Note:

When EN[0] has been set to 0, a delay occurs for Bit 0 to be read as 0 because disabling the I2C depends on I2C bus activities.

- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0x9C
- **Reset Value**: 0x00000000

Table 8-161 I2C Enable Status Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	S_RX_DATA_LOST	R	0x0	Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data

Bits	Field Name	RW	Reset	Description
				byte received from an I2C transfer due to the setting bit 0 of EN from 1 to 0. When this bit is read as 1, I2C is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. **Note**: If the remote I2C master terminates the transfer with a STOP condition before the I2C has a chance to NACK transfer, and EN[0] has been set to 0, then this bit is also set to 1. When this bit is read as 0, I2C is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. The CPU can safely read this bit when EN (bit 0) is read as 0. **Value**: • 0x0 (INACTIVE): Slave RX Data is not lost • 0x1 (ACTIVE): Slave RX Data is lost
1	S_DIS_BUSY	R	0x0	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting bit 0 of the EN register from 1 to 0. This bit is set when the CPU writes a 0 to the EN register while: I2C is receiving the address byte of the Slave-Transmitter operation from a remote master. Address and data bytes of the Slave-Receiver operation from a remote master. When this bit is read as 1, I2C is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the slave address set in I2C (S_ADDR register) or if the transfer is completed before EN is set to 0 but has not taken effect. **Note**: If the remote I2C master terminates the transfer with a STOP condition before the I2C has a chance to NACK a

Bits	Field Name	RW	Reset	Description
				transfer, and EN[0] has been set to 0, then this bit will also be set to 1. When this bit is read as 0, I2C is deemed to have been disabled when there is master activity, or when the I2C bus is idle. The CPU can safely read this bit when EN (bit 0) is read as 0. **Value**: • 0x0 (INACTIVE): Slave is disabled when it is idle • 0x1 (ACTIVE): Slave is disabled when it is active
0	EN	R	0x0	EN Status. This bit always reflects the value driven on the output port EN. When read as 1, I2C is deemed to be in an enabled state. When read as 0, I2C is deemed completely inactive. **Note**: The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read S_RX_DATA_LOST (bit 2) and S_DIS_BUSY (bit 1). **Value**: • 0x0 (DISABLED): I2C disabled • 0x1 (ENABLED): I2C enabled

8.4.3.4.40 FS_SPKLEN

- **Name**: I2C SS, FS or FM+ Spike Suppression Limit
- **Description**: This register is used to store the duration, measured in clock cycles, of the longest spike that is filtered out by the spike suppression logic when the component is operating in Standard Mode (SS), Fast Mode (FS), or Fast Plus (FM+) Mode. The relevant I2C requirement is tSP as detailed in the I2C Bus Specification. This register must be programmed with a minimum value of 1.
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0xA0
- **Reset Value**: 0x00000005

Table 8-162 I2C SS, FS or FM+ spike suppression limit

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
7:0	FS_SPKLEN	RW	0x5	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in clock cycles, of the longest spike in the SCL or SDA lines that will be filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled which corresponds to the EN[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than 1 being written, and any attempt to write a value less than 1 results in 1 being set.

8.4.3.4.41 HS_SPKLEN

- **Name**: I2C HS Spike Suppression Limit Register
- **Description**: This register is used to store the duration, measured in clock cycles, of the longest spike that is filtered out by the spike suppression logic when the component is operating in High-speed mode. The relevant I2C requirement is tSP as detailed in the I2C Bus Specification. This register must be programmed with a minimum value of 1 and is implemented only if the component is configured to support High-speed mode.
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0xA4
- **Reset Value**: 0x00000001

Table 8-163 I2C HS Spike Suppression Limit Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	HS_SPKLEN	RW	0x1	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in clock cycles, of the longest spike in the SCL or SDA lines that will be filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled which corresponds to the EN[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than 1 being written, and any attempt to write a value less than 1 results in 1 being set.

8.4.3.4.42 SCL_STUCK_AT_LOW_TIMEOUT

- **Name**: I2C SCL Stuck at Low Timeout Register
- **Description**: This register is used to store the duration, measured in ic_clk cycles, used to generate an interrupt (SCL_STUCK_AT_LOW) if SCL is held low for the SCL_STUCK_LOW_TIMEOUT duration
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0xAC
- **Reset Value**: 0xFFFFFFFF

Table 8-164 I2C SCL Stuck at Low Timeout Register

Bits	Field Name	RW	Reset	Description
31:0	SCL_STUCK_LOW_TIMEOUT	RW	0xFFFFFFFF	I2C generates the interrupt to indicate SCL stuck at low (SCL_STUCK_AT_LOW) if it detects that the SCL is stuck at low for the SCL_STUCK_LOW_TIMEOUT in units of ic_clk period. This register can be written only when the I2C interface is disabled which corresponds to the EN[0] register being set to 0. Writes at other times have no effect.

8.4.3.4.43 SDA_STUCK_AT_LOW_TIMEOUT

- **Name**: I2C SDA Stuck at Low Timeout Register
- **Description**: This register is used to store the duration, measured in ic_clk cycles, used to recover the data (SDA) line through sending SCL pulses if SDA is held low for the SDA_STUCK_LOW_TIMEOUT duration
- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0xB0
- **Reset Value**: 0xFFFFFFFF

Table 8-165 I2C SDA Stuck at Low Timeout Register

Bits	Field Name	RW	Reset	Description
31:0	SDA_STUCK_LOW_TIMEOUT	RW	0xFFFFFFFF	I2C initiates the recovery of SDA line through enabling the SDA_STUCK_RECOVERY_ENABLE (EN[3]) register bit, if it detects that the SDA is stuck at low for the SDA_STUCK_LOW_TIMEOUT in units of ic_clk period.

8.4.3.4.44 CLR_SCL_STUCK_DET

- **Name**: Clear SCL Stuck at Low Detect Interrupt Register
- **Description**: Clear SCL Stuck at Low Detect Interrupt Register

- **Base Address**: 0x4000C300 + x*0x100
- **Offset**: 0xB4
- **Reset Value**: 0x00000001

Table 8-166 I2C Clear SCL Stuck at Low Detect Interrupt Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLR_SCL_STUCK_DET	R	0x0	Read this register to clear the SCL_STUCK_AT_LOW interrupt (Bit 14) of the RAW_INT_STAT register.

8.4.3.5 Electrical Specifications

Table 8-167 Electrical Specifications

Symbol	Parameter	Standard Mode (SS)		Fast Mode (FS)		Fast-plus Mode (FM+)		High-speed Mode (HS)		Unit
		Min.	Max.	Min.	Max.	Min.	Max.	Min.	Max.	
f_{SCL}	SCL clock frequency	0	100	0	400	0	1000	0	2000	kHz
t_{LOW}	SCL clock LOW time	4.7	-	1.3	-	0.5	-	0.16	-	µs
t_{HIGH}	SCL clock HIGH time	4.0	-	0.6	-	0.26	-	0.06	-	µs
t_{RSDA}	SDArise time	-	1000	20	300	-	120	10	80	ns
t_{FSDA}	SDA fall time	-	300		300		120	10	80	ns
t_{RSCL}	SCLrise time	-	1000	20	300	-	120	10	40	ns
t_{FSCL}	SCL fall time	-	300		300		120	10	40	ns
t_{HD_STA}	START hold time	4.0	-	0.6	-	0.26	-	0.16	-	µs
t_{SU_DAT}	SDA setup time	250	-	100	-	50	-	10	-	ns
t_{HD_DAT}	SDA hold time	0	-	-	-	-	-	0	70	µs
t_{SU_STO}	STOP setup time	4.0	-	0.6	-	0.26	-	0.16	-	µs
t_{BUF}	BUS FREE time from STOP to START	4.7	-	1.3	-	0.5	-	-	-	µs

Note:

The bus load capacitance Cb = 100 pF.

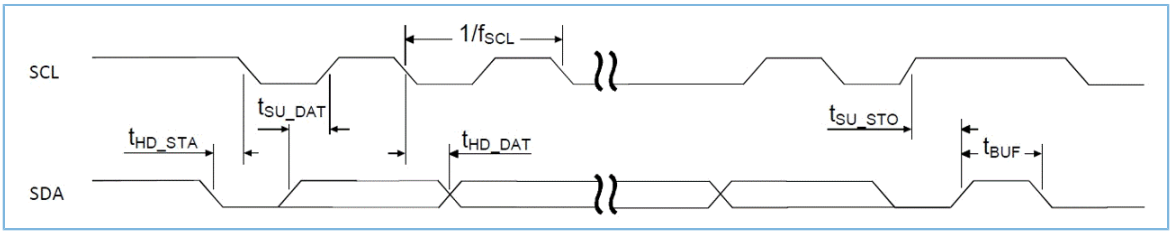

Figure 8-35 I2C timing diagram

8.4.4 USB 2.0

8.4.4.1 Introduction

GR5526 SoC provides a universal serial bus (USB) device controller that is compliant with USB 2.0 Full-Speed specification. The USB device controller supports six endpoints and four data transfer modes: control transfer, bulk transfer, interrupt transfer, and isochronous transfer.

8.4.4.2 Main Features

- Complaint with USB 2.0 Full-Speed specification
- Controllable pull-up/pull-down resistor at D+/D-
- Integrated USB PHY on GR5526 SoCs
- Support for six endpoints:
  - EP0 as a control endpoint that integrates IN and OUT
  - EP1 as an OUT endpoint supporting both interrupt and bulk transfers
  - EP2 and EP3 as IN endpoints supporting both interrupt and bulk transfers
  - EP4 as an IN endpoint supporting isochronous transfer
  - EP5 as an OUT endpoint supporting isochronous transfer
- Diverse FIFO sizes:
  - EP0: 2 x 64 bytes FIFOs
  - EP1, EP2, and EP3: 64 bytes FIFO for each
  - EP4 and EP5: 1023 bytes FIFO for each
- DMA (available for EP3, EP4, and EP5 only)

The USB system of GR5526 comprises four modules: a USB device controller (the light blue module in the Figure 8-36), a clock module (clock data recovery submodule and OSC clock generation source), an AON power module (5 V VBUS to 3.3 V LDO), and a USB transceiver module. The figure below shows the block diagram of a USB system.

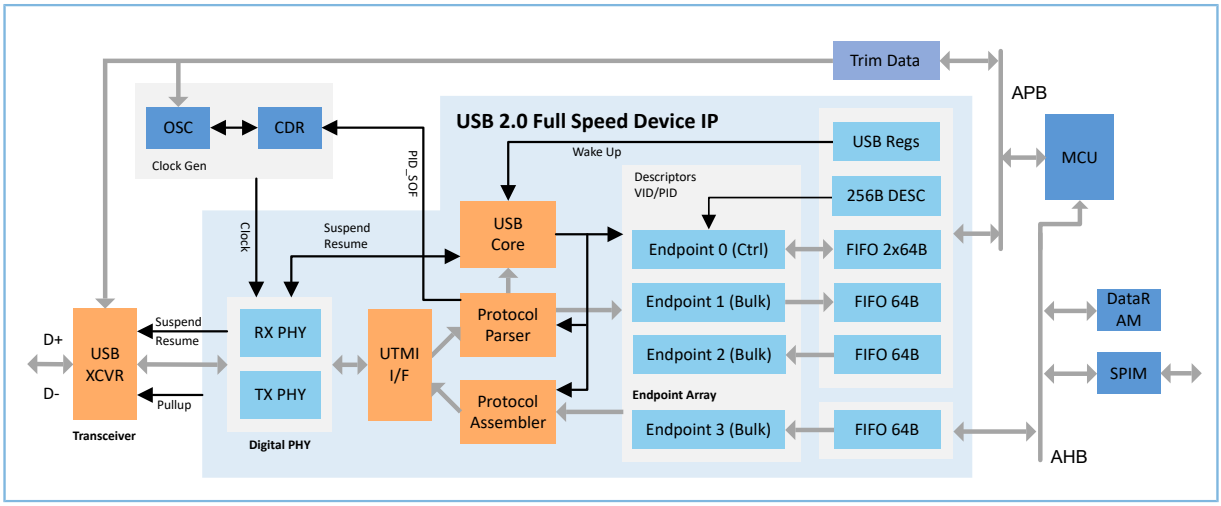

Figure 8-36 GR5526 USB system block diagram

The USB device controller module contains the following submodules:

- Digital PHY: receives differential signals from the USB transceiver (XCVR). The submodule generates parallel data and converts the data into differential data in NRZI code that will be then transmitted to the USB XCVR.
- UTMI I/F: converts parallel signals into standard USB 2.0 Transceiver Macrocell Interface (UTMI) signals.
- Protocol processing submodule: parses 8-bit parallel data signals received from the UTMI I/F and disassembles the data into different packet fields. The data is then transmitted to the UTMI I/F.
- Endpoint control submodule: controls write/read permissions of endpoints.
- Register submodule: receives control commands from MCU and returns the status and interrupt information of the USB device controller module.
- AON power submodule: detects connection/disconnection of a USB device to/from the host; converts 5 V VBUS supply from the host into 3.3 V supply for the USB XCVR.

8.4.4.3 Functional Description

8.4.4.3.1 Endpoint Configurations

Table 8-168 Endpoint configurations of GR5526 USB device controller

EP#	Direction	Type	FIFO Size	Functionality	Access	DMA
0	IN/OUT	Control	2 x 64 bytes	Enumeration/Config	APB	Not supported
1	OUT	Bulk/Interrupt	64 bytes	Data from host	APB	Not supported
2	IN	Bulk/Interrupt	64 bytes	Data to host	APB	Not supported
3	IN	Bulk/Interrupt	64 bytes	Data to host	AHB/APB	Supported
4	IN	Isochronous	1023 bytes	Data to host	AHB/APB	Supported
5	OUT	Isochronous	1023 bytes	Data from host	AHB/APB	Supported

The data transfer modes of EP1, EP2, and EP3 can be configured to either interrupt or bulk by the MCU through registers before USB reset. Default mode: bulk transfer.

EP0 serves as a standard control endpoint for USB enumeration. EP0 receives device requests from the host and returns descriptors; alternatively, EP0 can obtain configurations of diverse-type data sent out by the host.

EP1 receives data sent out from the host in bulk or interrupt mode. The data can be transferred in a packet of up to 64 bytes, and the data content is parsed by the MCU. After enabling EP1 interrupt transfer, the USB device controller parses data sent out from the host and informs the MCU of the packet size of the received data. The MCU reads related FIFO address to obtain the data.

EP2 sends out information to software on the host side and responds to IN requests that data should be sent to the host in bulk or interrupt mode. The data packet size is up to 64 bytes. The MCU writes to-be-transferred data to a corresponding FIFO address and informs the USB device controller to send out the data after packaging it into a packet. The packet format is defined by software.

In general, EP3 supports bulk transfer of large amounts of data. The EP3 accesses the system RAM in DMA mode as an AHB master. It transmits a large amount of data to the USB host. Alternatively, EP3 supports data write by the MCU(4 bytes align) through Advanced Peripheral Bus (APB) and transmits data to the USB host.

EP4 sends isochronous data to the host and supports a FIFO of 1023 bytes. It can be used to transmit audio data and respond to isochronous IN data requests from the host. Similar to EP3, EP4 supports data write by the MCU at the FIFO entry address through APB and transmits data to the USB host. In addition, EP4 accesses the system RAM in DMA mode as an AHB master.

EP5 receives isochronous data from the host and supports a FIFO of 1023 bytes. It can be used to transmit audio data and respond to isochronous OUT data requests from the host. EP5 enables the MCU to read data at the FIFO entry address through APB. In addition, EP5 accesses the system RAM in DMA mode as an AHB master.

8.4.4.3.2 USB Enumeration Methods

GR5526 supports USB enumeration by either hardware or software:

- By hardware: The built-in finite-state machine (FSM) automatically completes the USB enumeration, not requiring MCU or DMA.
- By software: The MCU parses and responds to all device requests while EP0 serves as a passthrough channel.

8.4.4.3.2.1 Automatic Hardware-enabled Enumeration

In hardware-enabled enumeration, the built-in FSM controls the enumeration procedures, and EP0 automatically parses and responds to standard device requests sent by the host. The hardware-enabled enumeration features highest enumeration efficiency, fast response, and no timeout. In addition, it requires no support from the MCU and exerts the least impact on system. However, this enumeration approach is less flexible to use compared with software-enabled enumeration (supporting enumeration of some USB devices only).

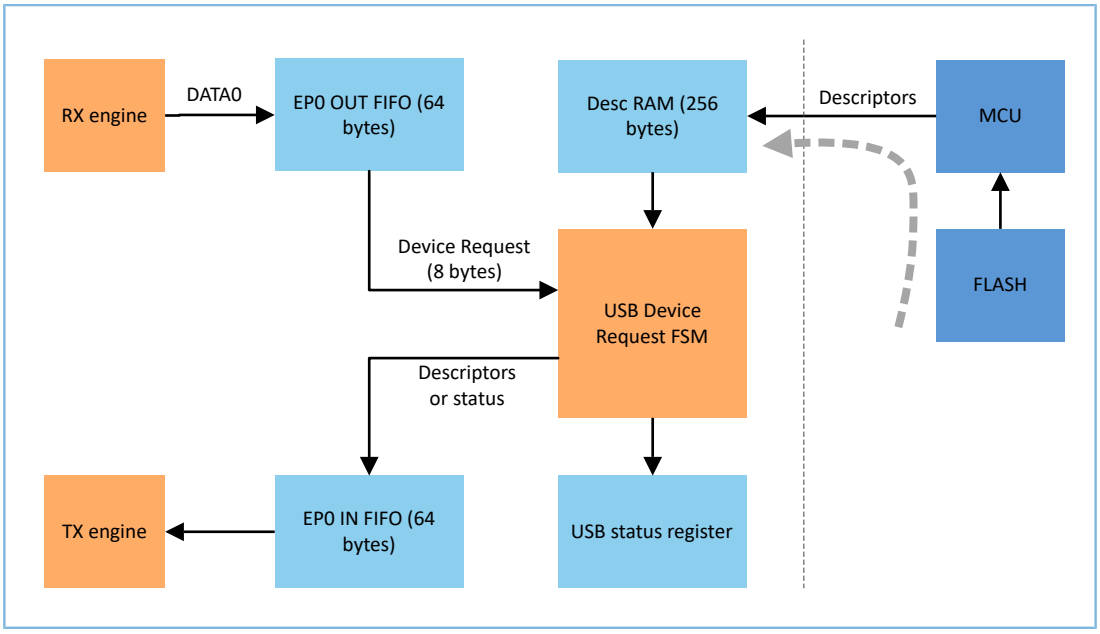

Figure 8-37 Hardware-enabled enumeration diagram

Descriptors required in hardware-enumeration such as vendor IDs (VID) and product IDs (PID) of USB devices are stored in GR5526 Flash. The hardware enumeration module provides a built-in 256-byte RAM to store enumeration data. Power on the GR5526 SoC, and the MCU writes the descriptors to the RAM through APB in Flash. During enumeration, the FSM reads corresponding descriptors in the RAM and then responds to device requests sent from the host.

8.4.4.3.2.2 Software-enabled Enumeration

In software-enabled enumeration, EP0 serves as a bidirectional data transfer channel, and the USB device controller module does not parse payloads. This makes the software-enabled enumeration flexible to use (supporting enumeration of all USB devices) and allows users to modify MCU firmware files to ensure optimal interoperability.

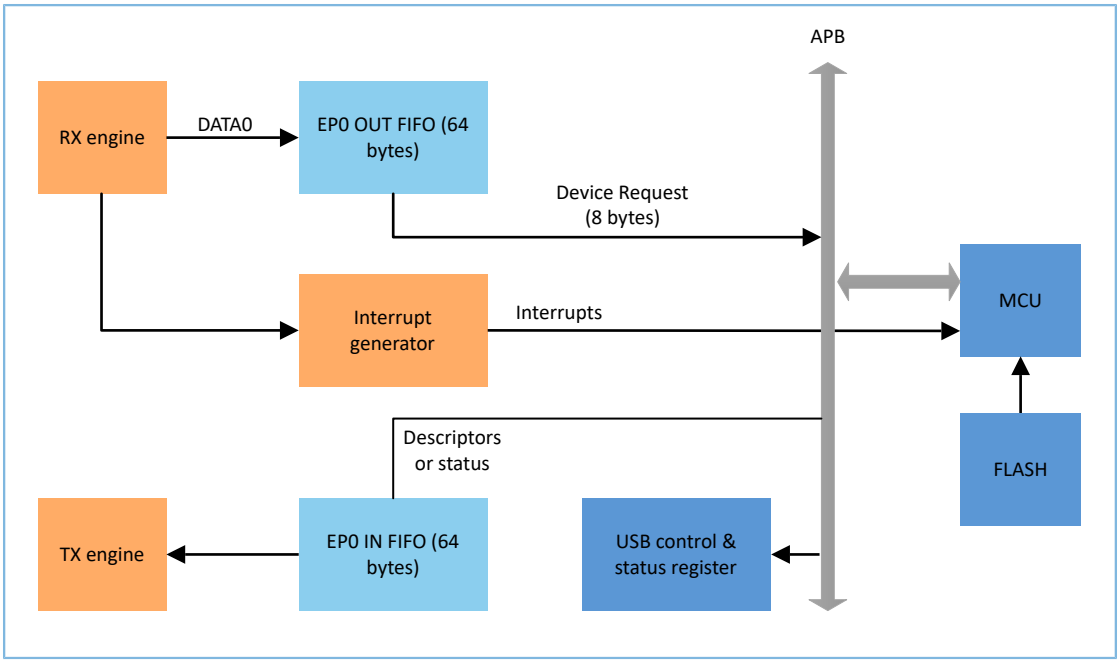

Figure 8-38 Software-enabled enumeration diagram

Software-enabled enumeration steps:

1. EP0 sends out an interrupt signal to the MCU after receiving a complete data packet (when an end of packet or EOP signal is detected and CRC validation).
2. The MCU processes the interrupt signal and then reads and parses data bytes one after another from the EP0 OUT FIFO (the first byte of a packet indicates the packet size).
3. The MCU successively writes the corresponding data including descriptors that are read from Flash into the IN FIFO of EP0 according to the request type.
4. When data write completes, set the EP0_1_CTRL register to EP0_DATA_READY by software to inform the USB device controller that the complete data packet has been written to the IN FIFO of EP0.
5. EP0 sends the data in the IN FIFO to the USB host in a packet.

8.4.4.3.3 Bulk and Interrupt Transfer

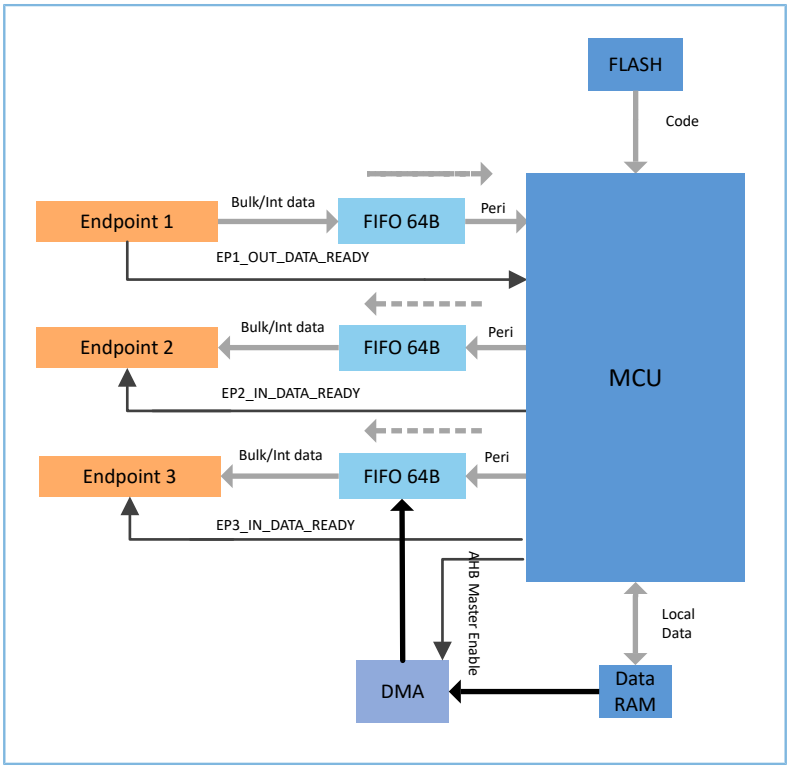

Figure 8-39 EP1, EP2, EP3 data transfer diagram

After EP1 is enabled, it receives the OUT packet sent by the host based on the configured data transfer type (bulk or interrupt), and parses the payload size in the packet as well as contents saved in its FIFO. EP1 then generates the EP1_OUT_READY interrupt to notify the MCU to read payload of the same size from the FIFO.

When the MCU needs to send messages (including data in response to command(s) to EP1, and the status agreed between the MCU software and the host, such as an interrupt status) to the host, the action should be first permitted by EP2. EP1, EP2, and EP3 are in bulk transfer mode by default. If a specific latency is required, users can configure EP2 to interrupt mode via the MCU prior to data transfer.

During EP2 data transfer:

1. The MCU checks the status register EP2_DATA_READY via APB.
2. If the space in EP2 FIFO is sufficient, the MCU writes the values read from the register in sequence to the EP2 FIFO in the agreed format. Otherwise, wait until the space becomes sufficient.
3. When data write completes, the MCU writes to the EP2_DATA_READY register via APB, to inform USB device controller of data write completion and data ready for transmission, to avoid incomplete data transmission caused by data transfer before data write to FIFO completes.
4. When the USB module receives EP2 IN token from the host, it packs and sends the data from the FIFO to the host.

In addition to the functions of EP2, EP3 supports DMA, which enables bulk data transfer from the MCU to the host: Set the allocation address on RAM for DMA transfer and the size of data to be transferred, and enable the DMA function. By default, the USB module allocates data to the EP3 FIFO before data transfer.

1. The MCU configures USB registers via APB. Set the condition to trigger EP3 interrupt is when the available space in EP3 FIFO equal to or above 64 bytes.
2. The MCU checks whether there are sufficient data for USB data transfer based on DMA interrupt. If no, wait until the data blocks become sufficient; if yes, ask USB host via EP2 to fetch data and then proceed to the next step.
3. The MCU configures the AHB master interface module of EP3 [the configuration covers SRAM offset address, total size of data to be transferred, and data size (burst size) in one read operation by AHB].
4. AHB master transfers data from the designated address in SRAM to FIFO in the size of one read operation as configured.
5. After receiving IN Token from the host, EP3 packs and sends data to the host; if the data transfer (one frame) has not completed, AHB master will continue to transfer data from SRAM until the data transfer completes.

The host software parses messages; USB host only passes through data, and places no limit to the payload format and content.

8.4.4.3.4 Isochronous Transfer

EP4 and EP5 only support isochronous data transfer, and each uses its FIFO (1023 bytes) for buffer.

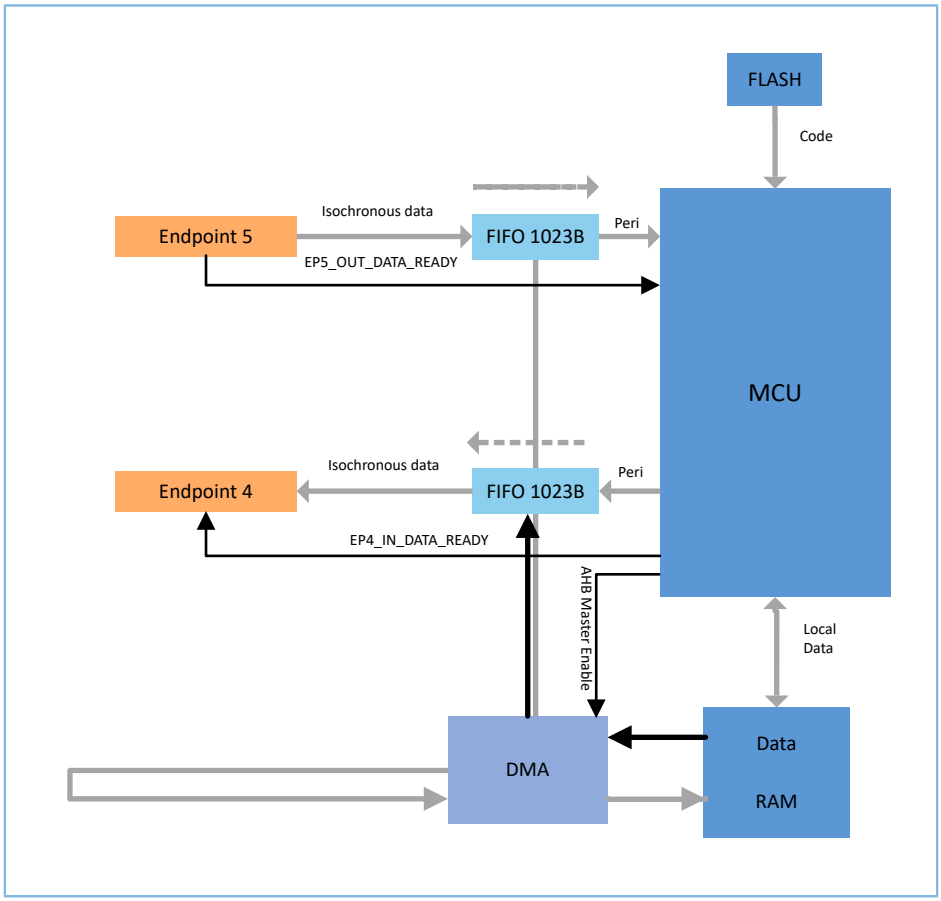

Figure 8-40 EP4 and EP5 data transfer diagram

During EP4 data transfer:

- When EP4 sends isochronous data to the host via FIFO:
  1. Enable TX interrupt of EP4 FIFO and the empty packet function of EP4 (to return a packet with null data before data is ready), and disable the pending state of EP4.
  2. Query whether EP4 FIFO is sending data; clear data in FIFO if data TX completes.
  3. Write data to EP4 in every 4 bytes. If the last data segment written to EP4 does not reach 4 bytes, re-configure the size for the last data transfer of EP4 FIFO based on the actual data size (1 byte, 2 bytes, or 3 bytes).
  4. Set the data transfer status of EP4 FIFO to complete.
- When EP4 sends isochronous data to the host via DMA:
  1. Enable DMA TX interrupt and the empty packet function of EP4 (to return a packet with null data before data is ready), and disable the pending state of EP4.
  2. Query whether EP4 is sending data via DMA; clear data in FIFO if DMA data TX completes.
  3. Set DMA data TX size of EP4 and the RAM entry address to receive data via DMA.
  4. Enable DMA data TX of EP4 as AHB master.

During EP5 data transfer:

- When EP4 receives isochronous data from the host via FIFO:
  1. Enable RX interrupt of EP5, and disable the pending state of EP5.
  2. After receiving the OUT data from the host, EP5 parses the received payload and writes the parsed data to FIFO; it then generates an interrupt EP5_OUT_READY to inform the MCU of the received data size.
  3. The MCU reads data in accordance with the received data size from the entry address of the EP5 FIFO.
- When EP5 receives isochronous data from the host via DMA:
  1. Configure the RAM entry address for data transferred via DMA, and the data size to generate interrupt to indicate EP5 DMA RX completes.
  2. Enable DMA RX interrupt and DMA RX timeout interrupt of EP5, and set the timeout period: If the size of received data does not equal the size to generate an interrupt, an interrupt will be generated after the system times out. Disable the pending state of EP5.
  3. After receiving the OUT data from the host, EP5 parses the received payload and writes the parsed data to FIFO. Data will be transferred to the RAM address as configured via DMA; data transfer ends when the transferred size reaches the pre-set size, and an interrupt will be generated to inform the MCU of DMA data RX completion.
  4. If the transferred data size does not reach or exceed the size to generate an interrupt, an interrupt will be generated after the system times out, to inform the MCU of data transfer completion.

8.4.4.4 Registers

8.4.4.4.1 CTRL

Name: USB Control Register

Description: This register controls USB base state.

Base Address: 0x0400A900

Offset: 0x0

Reset Value: 0x00000000

Table 8-169 USB Control Register

Bits	Field Name	RW	Reset	Description
31:18	RSVD	R		Reserved bits
17	EP5_OUT_DATA_READY	RW	0x0	OUT data in EP5 is ready. The MCU will clear this bit after reading the data.
16	EP0_FIFO_SWITCH_EN	RW	0x0	This bit is used when MCU_ENUM = 0. When hardware enumeration is finished, disable enumeration by hardware.
15	EP1_OUT_DATA_READY	RW	0x0	OUT data in EP1 is ready. The MCU will clear this bit after reading the data. The USB will issue NAK before the data is read out.
14:8	FUNC_ADDR	RW	0x0	USB function address. If taking enumeration, the MCU fills in the address; otherwise this field is automatically set by the USB device controller.
7	CMD_ERR	W	0x0	Error occurs when the MCU interprets the USB device request. 1: Command or OUT data is wrong. 0: No error.
6	CMD_OK	W	0x0	The MCU interprets the request sent by the host to the USB device. • If there is no OUT data in the command, the MCU will set the bit after interpreting the command. • If there is OUT data in the command, the MCU will set the bit after reading the OUT data packet.
5	CFG_STATUS	RW	0x0	USB is in configured status: If taking enumeration, the MCU fills in the bit; otherwise this field is automatically set by the USB device controller.
4	ADDR_STATUS	RW	0x0	USB is in configuring address status. If taking enumeration, the MCU fills in the bit; otherwise this field is automatically set by the USB device controller.

Bits	Field Name	RW	Reset	Description
3	DEV_REMOTE_WAKEUP	RW	0x0	REMOTE_WAKEUP feature. If taking enumeration, the MCU sets this bit for USB device; otherwise this bit is automatically set by the USB device controller.
2	MCU_WAKEUP	W	0x0	The MCU wakes up USB host remotely. When the MCU writes 1 to this bit, and "REMOTE_WAKEUP" feature is set by the host, the USB will wake up the host. This bit will be cleared after USB wakeup.
1	EP0_OUT_DATA_READY	RW	0x0	OUT data in EP0 is ready. The MCU will clear this bit after reading the data. USB will issue NAK before the data is read out.
0	MCU_ENUM	RW	0x0	The type of the MCU taking the enumeration process: 0: Hardware-enabled enumeration 1: Software-enabled enumeration (by MCU)

8.4.4.4.2 EP0_1_CTRL

Name: USB EP0 and EP1 Control Register

Description: This register controls USB EP0 and EP1.

Base Address: 0x0400A900

Offset: 0x04

Reset Value: 0x00000000

Table 8-170 USB EP0 and EP1 Control Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	EP1_OFIFO_CLR	W	0x0	Write 1 to this bit to clear EP1 OUT FIFO.
1	EP0_IFIFO_CLR	W	0x0	Write 1 to this bit to clear EP0 IN FIFO.
0	EP0_DATA_READY	RW	0x0	The bit is set by MCU when MCU has sent all data to EP0 IN FIFO. The bit is set by USB device controller when IN FIFO has max-packet-size data to transfer. After USB device controller sends all EP0 data, it will clear this bit. MCU shall poll this bit before send data to EP0 FIFO, and cannot write data until this bit is cleared.

8.4.4.4.3 EP2_CTRL

Name: USB EP2 Control Register

Description: This register controls USB EP2.

Base Address: 0x0400A900

Offset: 0x08

Reset Value: 0x00000000

Table 8-171 USB EP2 Control Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	EP2_IFIFO_CLR	W	0x0	Write 1 to this bit to clear EP2 IN FIFO.
0	EP2_DATA_READY	RW	0x0	The bit is set by MCU when MCU has sent all data to EP2 IN FIFO. The bit is set by USB device controller when IN FIFO has max-packet-size data to transfer. After USB device controller sends all EP2 data, it will clear this bit. MCU shall poll this bit before sending data to EP2 FIFO, and cannot write data until this bit is cleared.

8.4.4.4.4 EP3_CTRL

Name: USB EP3 Control Register

Description: This register controls USB EP3.

Base Address: 0x0400A900

Offset: 0x0C

Reset Value: 0x00000000

Table 8-172 USB EP3 control Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	EP3_IFIFO_CLR	W	0x0	Write 1 to this bit to clear EP3 IN FIFO.
0	EP3_DATA_READY	RW	0x0	The bit is set by MCU when MCU has sent all data to EP3 IN FIFO. The bit is set by USB controller when IN FIFO has max-packet-size data to transfer.

Bits	Field Name	RW	Reset	Description
				After USB device controller sends all EP3 data, it will clear this bit. MCU shall poll this bit before send data to EP3 FIFO, and cannot write data until this bit is cleared.

8.4.4.4.5 EP_ATTR

Name: USB Set Endpoint Attribute Register

Description: This register sets USB EP1, EP2 and EP3 attribute.

Base Address: 0x0400A900

Offset: 0x10

Reset Value: 0x00000202

Table 8-173 USB Set Endpoint Attribute Register

Bits	Field Name	RW	Reset	Description
31:11	RSVD	R		Reserved bits
10	EP3_HALT_MCU	RW	0x0	Set this bit to halt EP3.
8:9	EP3_ATTR	RW	0x2	EP3 attribute setting: 00: Interrupt 01: Isochronous 10: Bulk 11: Undefined MCU must keep this data consistent with the content of descriptor.
7	RSVD	R		Reserved bits
6	EP2_HALT_MCU	RW	0x0	Set this bit to halt EP2.
4:5	EP2_ATTR	RW	0x0	EP2 attribute setting: 00: Interrupt 01: Isochronous 10: Bulk 11: Undefined MCU must keep this data consistent with the content of descriptor.
3	RSVD	R		Reserved bits
2	EP1_HALT_MCU	RW	0x0	Set this bit to halt EP1
0:1	EP1_ATTR	RW	0x2	EP1 attribute setting: 00: Interrupt

Bits	Field Name	RW	Reset	Description
				01: Isochronous 10: Bulk 11: Undefined MCU must keep this data consistent with the content of descriptor.

8.4.4.4.6 INT_STAT

Name: USB Interrupt Status Register

Description: This register returns the interrupt status of USB.

Base Address: 0x0400A900

Offset: 0x14

Reset Value: 0x00000000

Table 8-174 USB Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:26	RSVD	R		Reserved bits
25	EP5_TIMER_OUT_ERR	R	0x0	AHB master EP5 transmission timeout error.
24	EP5_AHB_XFER_DONE	R	0x0	AHB master EP5 transfer done.
23	EP4_DAT_SENT	R	0x0	The data in the EP4 IN FIFO has been sent to the host.
22	EP4_AHB_XFER_DONE	R	0x0	AHB master EP4 transfer done.
21	EP5_OUT_READY	R	0x0	The data in the EP5 from the USB host is ready.
20	INTO_CONFIG	R	0x0	The hardware enters the enumeration status, and is valid when MCU_ENUM = 0.
19	EP3_DAT_SENT	R	0x0	The data in the EP3 IN FIFO has been sent to the host.
18	EP2_DAT_SENT	R	0x0	The data in the EP2 IN FIFO has been sent to the host.
17	EP0_DAT_SENT	R	0x0	The data in the EP0 IN FIFO has been sent to the host.
16	SOF	R	0x0	SOF interrupt.
15	BYTE_ERR	R	0x0	USB byte error.
14	BIT_STUFF_ERR	R	0x0	USB bit stuff error.
13	SYNC_ERR	R	0x0	USB SYNC error.
12	AHB_EP3_XFER_DONE	R	0x0	AHB master EP3transfer done.
11	NSE_ERR	R	0x0	No such endpoint error.
10	AHB_EP3_XFER_ERR	R	0x0	AHB master EP3 receives error response.
9	HOST_RESET	R	0x0	USB host reset.
8	PID_CRC_ERR	R	0x0	PID CRC error.

Bits	Field Name	RW	Reset	Description
7	PID_CKS_ERR	R	0x0	PID checksum error.
6	SEQ_ERR	R	0x0	DATA0/DATA1 PID sequence error.
5	TO_ERR	R	0x0	RX/TX timeout error.
4	UPID_ERR	R	0x0	Receive unsupported PID error.
3	CRC16_ERR	R	0x0	Receive CRC error data.
2	EP1_OUT_READY	R	0x0	The data in the EP1 from the USB host is ready.
1	EP0_OUT_READY	R	0x0	The setup data in the EP0 from the USB host is ready.
0	USB_SUSPEND	R	0x0	MCU takes the enumeration process. USB device controller will turn to the suspended status.

8.4.4.4.7 INT_EN

Name: USB Interrupt Enable Register

Description: This register enables USB interrupts.

Base Address: 0x0400A900

Offset: 0x18

Reset Value: 0x003EFDFF

Table 8-175 USB Interrupt Enable Register

Bits	Field Name	RW	Reset	Description
31:26	RSVD	R		Reserved bits
25	EP5_TIMER_OUT_ERR_EN	RW	0x01	Enable EP5 transfer timeout interrupt.
24	EP5_AHB_XFER_DONE_EN	RW	0x01	Enable AHB master EP5 transfer done interrupt.
23	EP4_DAT_SENT_EN	RW	0x01	Enable the interrupt source: Data in the EP4 IN FIFO has been sent to the host.
22	EP4_AHB_XFER_DONE_EN	RW	0x01	Enable AHB master EP4 transfer done interrupt.
21	EP5_OUT_READY_EN	RW	0x01	Enable the interrupt source: Data in endpoint5 from USB host is ready.
20	INTO_CONFIG_EN	RW	0x0	Enable the interrupt that the hardware completes enumeration process and enters the configured status, which is valid when EP0_FIFO_SWITCH_EN = 1.
19	EP3_DAT_SENT_EN	RW	0x01	Enable the interrupt that EP3 IN FIFO data has been sent to the host.
18	EP2_DAT_SENT_EN	RW	0x0	Enable the interrupt that EP2 IN FIFO data has been sent to the host.

Bits	Field Name	RW	Reset	Description
17	EP0_DAT_SENT_EN	RW	0x01	Enable the interrupt that EP0 IN FIFO data has been sent to host.
16	SOF_EN	RW	0x01	Enable the SOF interrupt.
15	BYTE_ERR_EN	RW	0x01	Enable the USB byte error interrupt.
14	BIT_STUFF_ERR_EN	RW	0x01	Enable the USB bit stuff error interrupt.
13	SYNC_ERR_EN	RW	0x0	Enable the USB SYNC error interrupt.
12	AHB_EP3_XFER_DONE_EN	RW	0x01	Enable the AHB master EP3 transfer done interrupt.
11	NSE_ERR_EN	RW	0x01	Enable the no such endpoint error interrupt.
10	AHB_EP3_XFER_ERR_EN	RW	0x01	Enable the interrupt source: AHB master EP3 receives ERROR response.
9	HOST_RESET_EN	RW	0x01	Enable USB host reset interrupt.
8	PID_CRC_ERR_EN	RW	0x01	Enable PID CRC error interrupt.
7	PID_CKS_ERR_EN	RW	0x01	Enable PID checksum error interrupt.
6	SEQ_ERR_EN	RW	0x01	Enable DATA0/DATA1 PID sequence error interrupt.
5	TO_ERR_EN	RW	0x01	Enable RX/TX timeout error interrupt.
4	UPID_ERR_EN	RW	0x01	Enable receiving unsupported PID interrupt.
3	CRC16_ERR_EN	RW	0x01	Enable receiving CRC error data interrupt.
2	EP1_OUT_READY_EN	RW	0x01	Enable the interrupt that the data in EP1 from USB host is ready.
1	EP0_OUT_READY_EN	RW	0x01	Enable the interrupt that setup data in EP0 from USB host is ready.
0	USB_SUSPEND_EN	RW	0x01	Enable the interrupt that MCU takes the enumeration process of USB controller turning to suspended status.

8.4.4.4.8 INT_CLR

Name: USB Interrupt Clear Register

Description: This register clears USB interrupts.

Base Address: 0x0400A900

Offset: 0x1C

Reset Value: 0x00000000

Table 8-176 USB interrupt clear Register

Bits	Field Name	RW	Reset	Description
31:26	RSVD	R		Reserved bits
25	EP5_TIMER_OUT_ERR_CLR	W	0x0	Clear EP5 transfer timeout interrupt.

Bits	Field Name	RW	Reset	Description
24	EP5_AHB_XFER_DONE_CLR	W	0x0	Clear AHB master EP5 transfer done interrupt.
23	EP4_DAT_SENT_CLR	W	0x0	Clear the interrupt source: Data in the EP4 IN FIFO has been sent to the host.
22	EP4_AHB_XFER_DONE_CLR	W	0x0	Clear AHB master EP4 transfer done interrupt.
21	EP5_OUT_READY_CLR	W	0x0	Clear the interrupt source: Data in endpoint5 from USB host is ready.
20	INTO_CONFIG_CLR	W	0x0	Clear the interrupt source: The hardware completes enumeration process and enters the configured status, which is valid when EP0_FIFO_SWITCH_EN = 1.
19	EP3_DAT_SENT_CLR	W	0x0	Clear the interrupt source: EP3 IN FIFO data has been sent to the host.
18	EP2_DAT_SENT_CLR	W	0x0	Clear the interrupt source: EP2 IN FIFO data has been sent to the host.
17	EP0_DAT_SENT_CLR	W	0x0	Clear the interrupt source: EP0 IN FIFO data has been sent to host.
16	SOF_CLR	W	0x0	Clear the SOF interrupt.
15	BYTE_ERR_CLR	W	0x0	Clear the USB byte error interrupt.
14	BIT_STUFF_ERR_CLR	W	0x0	Clear the USB bit stuff error interrupt.
13	SYNC_ERR_CLR	W	0x0	Clear the USB SYNC error interrupt.
12	AHB_EP3_XFER_DONE_CLR	W	0x0	Clear the AHB master EP3 transfer done interrupt.
11	NSE_ERR_CLR	W	0x0	Clear the no such endpoint error interrupt.
10	AHB_EP3_XFER_ERR_CLR	W	0x0	Clear the interrupt source: AHB master EP3 receives error response.
9	HOST_RESET_CLR	W	0x0	Clear USB host reset interrupt.
8	PID_CRC_ERR_CLR	W	0x0	Clear PID CRC error interrupt.
7	PID_CKS_ERR_CLR	W	0x0	Clear PID checksum error interrupt.
6	SEQ_ERR_CLR	W	0x0	Clear DATA0/DATA1 PID sequence error interrupt.
5	TO_ERR_CLR	W	0x0	Clear RX/TX timeout error interrupt.
4	UPID_ERR_CLR	W	0x0	Clear receiving unsupported PID interrupt.
3	CRC16_ERR_CLR	W	0x0	Clear receiving CRC error data interrupt.
2	EP1_OUT_READ_CLR Y	W	0x0	Clear the interrupt source: Data in EP1 from USB host is ready.
1	EP0_OUT_READY_CLR	W	0x0	Clear the interrupt source: Setup data in EP0 from USB host is ready.

Bits	Field Name	RW	Reset	Description
0	USB_SUSPEND_CLR	W	0x0	Clear the interrupt source: MCU takes the enumeration process of USB device controller turning to suspended status.

8.4.4.4.9 EP3_AHBM_RADDR

Name: AHB Master EP3 Reads Start Address Register

Description: This register enables the AHB master EP3 to read the start address.

Base Address: 0x0400A900

Offset: 0x20

Reset Value: 0x00000000

Table 8-177 AHB Master EP3 Reads Start Address Register

Bits	Field Name	RW	Reset	Description
31:0	EP3_RD_SADDR	RW	0x0	AHB master EP3 reads start address

8.4.4.4.10 EP3_AHMB_CTRL

Name: AHB Master EP3 Reads Control Register

Description: This register controls AHB master EP3.

Base Address: 0x0400A900

Offset: 0x24

Reset Value: 0x00004000

Table 8-178 AHB Master EP3 Reads Control Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:8	EP3_AHBM_BURST_SIZE	RW	0x40	AHB master EP3 burst length; default 64 bytes.
7:1	RSVD	R		Reserved bits
0	EP3_AHBM_EN	RW	0x0	Enable AHB master EP3.

8.4.4.4.11 CTRL0

Name: USB Control 0 Register

Description: This register controls USB PHY.

Base Address: 0x0400A900

Offset: 0x28

Reset Value: 0x00000048

Table 8-179 USB Control 0 Register

Bits	Field Name	RW	Reset	Description
31:19	RSVD	R		Reserved bits
18:16	PROBE_SEL	RW	0x0	Probe selection signal: • 0: Main protocol state machine ◦ 1: IDLE ◦ 2: TOKEN ◦ 3: IN ◦ 4: IN2 ◦ 5: OUT ◦ 6: OUT2A ◦ 7: OUT2B ◦ 8: UPDATEW ◦ 9: UPDATE ◦ 10: UPDATE2 • 1: RX state machine ◦ 1: IDLE ◦ 2: ACTIVE ◦ 4: TOKEN ◦ 8: DATA • 2: UTMI signals: RxValid, RxActive, TxValid, and TxReady • 3: SYNC state machine, and RxError ◦ 0: IDLE ◦ 1: K1 ◦ 2: J1 ◦ 3: K2 ◦ 4: J2 ◦ 5: K3 ◦ 6: J3 ◦ 7: K4 • 4: TX state machine ◦ 1: IDLE ◦ 2: DATA ◦ 4: CRC1

Bits	Field Name	RW	Reset	Description
				◦ 8: CRC2 • 5: DPLL state machine
15	RXD	R	0x0	RXD real time status.
14	INPUT_EDIAN_CTRL	RW	0x0	1: Data is in big endian. 0: Data is in small endian.
13	OUTPUT_ENDIAN_CTRL	RW	0x0	1: Data is in big endian. 0: Data is in small endian.
12:10	RSVD	R		Reserved bits
9	XCVR_DM_RPD_EN	RW	0x0	Set this bit to drive DM to pull down in transceiver test mode.
8	XCVR_DP_RPD_EN	RW	0x0	Set this bit to drive Data plus (D+) to pull down in transceiver test mode.
7	XCVR_DM_RPUSW_EN	RW	0x0	Set this bit to drive DM software to pull up in transceiver test mode.
6	XCVR_DP_RPUSW_EN	RW	0x01	Set this bit to drive DP SW to pull up in transceiver test mode. In function mode, this bit is driven by USB controller and cannot be set.
5	XCVR_DM_RPU_EN	RW	0x0	Set this bit to drive DM to pull up in transceiver test mode.
4	XCVR_DP_RPU_EN	RW	0x0	Set this bit to drive DP to pull up in transceiver test mode. In function mode, this bit is driven by USB controller and cannot be set.
3	XCVR_OEB	RW	0x01	Drive output enable buffer (OEB) in transceiver test mode.
2	DRIVE_DM	RW	0x0	Drive DM in transceiver test mode. Read this bit to get the DM status.
1	DRIVE_DP	RW	0x0	Drive DP in transceiver test mode. Read this bit to get the DP status.
0	TEST_MODE	RW	0x0	0: Function mode 1: USB transceiver test mode

8.4.4.4.12 EP3_XFER_LEN

Name: USB Total EP3 Transfer Length Register

Description: This register sets the total USB EP3 transfer length.

Base Address: 0x0400A900

Offset: 0x2C

Reset Value: 0x00000000

Table 8-180 USB Total EP3 Transfer Length Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	EP3_XFER_LEN	RW	0x0	Total USB EP3 transfer length. MCU should set this field for AHB transfer.

8.4.4.4.13 RX_CNT

Name: USB Received Data Sum Register

Description: This register gets received data sum in the EP0 FIFO and EP1 FIFO.

Base Address: 0x0400A900

Offset: 0x30

Reset Value: 0x00000000

Table 8-181 USB Received Data Sum Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:8	EP1_RX_DATA_SUM	R	0x0	USB receives data sum in the EP1 FIFO.
7:0	EP0_RX_DATA_SUM	R	0x0	USB receives data sum in the EP0 FIFO.

8.4.4.4.14 CFG_DESC_CTRL

Name: USB Config Descriptor Control Register

Description: This register controls USB config descriptor in hardware-enabled enumeration.

Base Address: 0x0400A900

Offset: 0x34

Reset Value: 0x00004912

Table 8-182 USB Config Descriptor Control Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:8	CFG_DESC_SIZE	RW	0x49	Config descriptor size.
7:0	CFG_DESC_START	RW	0x12	Config descriptor start address.

8.4.4.4.15 STR_DESC0_CTRL

Name: USB Language ID Descriptor Control Register

Description: This register controls USB language ID descriptor in hardware-enabled enumeration.

Base Address: 0x0400A900

Offset: 0x38

Reset Value: 0x00000470

Table 8-183 USB Language ID Descriptor Control Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:8	STR_DESC0_SIZE	RW	0x4	Language ID descriptor size
7:0	STR_DESC0_START	RW	0x70	Language ID descriptor start address

8.4.4.4.16 STR_DESC1_CTRL

Name: USB String Descriptor Control Register

Description: This register controls USB string descriptor in hardware-enabled enumeration.

Base Address: 0x0400A900

Offset: 0x3C

Reset Value: 0x00001290

Table 8-184 USB String Descriptor Control Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:8	STR_DESC1_SIZE	RW	0x12	String descriptor size
7:0	STR_DESC1_START	RW	0x90	String descriptor start address

8.4.4.4.17 EP0_FIFO_ADDR

Name: USB EP0 FIFO Address Register

Description: This register reads or sends USB EP0 FIFO data.

Base Address: 0x0400A900

Offset: 0x40

Reset Value: 0x00000000

Table 8-185 USB EP0 FIFO Address Register

Bits	Field Name	RW	Reset	Description
31:0	USB_EP0_FIFO	RW	0x0	USB EP0 FIFO address.

8.4.4.4.18 EP1_FIFO_ADDR

Name: USB EP1 FIFO Address Register

Description: This register reads USB EP1 FIFO data.

Base Address: 0x0400A900

Offset: 0x44

Reset Value: 0x00000000

Table 8-186 USB EP1 FIFO Address Register

Bits	Field Name	RW	Reset	Description
31:0	USB_EP1_FIFO	R	0x0	USB EP1 FIFO address.

8.4.4.4.19 EP2_FIFO_ADDR

Name: USB EP2 FIFO Address Register

Description: This register sends data to USB EP2 FIFO.

Base Address: 0x0400A900

Offset: 0x48

Reset Value: 0x00000000

Table 8-187 USB EP2 FIFO Address Register

Bits	Field Name	RW	Reset	Description
31:0	USB_EP2_FIFO	W	0x0	USB EP2 FIFO address.

8.4.4.4.20 EP3_FIFO_ADDR

Name: USB EP3 FIFO Address Register

Description: This register sends data to USB EP3 FIFO.

Base Address: 0x0400A900

Offset: 0x4C

Reset Value: 0x00000000

Table 8-188 USB EP3 FIFO Address Register

Bits	Field Name	RW	Reset	Description
31:0	USB_EP3_FIFO	W	0x0	USB EP3 FIFO address.

8.4.4.4.21 SRAM_ADDR

Name: USB Descriptor SRAM Address Register

Description: This register indicates USB descriptor SRAM address in hardware-enabled enumeration.

Base Address: 0x0400A900

Offset: 0x50

Reset Value: 0x00000000

Table 8-189 USB Descriptor SRAM Address Register

Bits	Field Name	RW	Reset	Description
31:0	USB_DESC_SRAM	W	0x0	USB descriptor SRAM address

8.4.4.4.22 STR_DESC2_CTRL

Name: USB String2 Descriptor Control Register

Description: This register controls USB string2 descriptor in hardware-enabled enumeration.

Base Address: 0x0400A900

Offset: 0x54

Reset Value: 0x00000A60

Table 8-190 USB String2 Descriptor Control Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:8	STR_DESC2_SIZE	RW	0x0A	When wValue[3:0] = 4'h2, string descriptor size (for string descriptor) should be reserved.
7:0	STR_DESC2_START	RW	0x60	When wValue[3:0] = 4'h2, descriptor start address (for string descriptor) should be reserved.

8.4.4.4.23 STR_DESC3_CTRL

Name: USB String3 Descriptor Control Register

Description: This register controls USB string3 descriptor in hardware-enabled enumeration.

Base Address: 0x0400A900

Offset: 0x58

Reset Value: 0x00000A70

Table 8-191 USB String3 Descriptor Control Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:8	STR_DESC3_SIZE	RW	0x0A	When wValue[3:0] = 4'h3, string descriptor size (for string descriptor) should be reserved.
7:0	STR_DESC3_START	RW	0x70	When wValue[3:0] = 4'h3, descriptor start address(for string descriptor) should be reserved.

8.4.4.4.24 EP4_CTRL

Name: USB EP4 Control Register

Description: This register controls USB EP4.

Base Address: 0x0400A900

Offset: 0x60

Reset Value: 0x00000000

Table 8-192 USB EP4 Control Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	EP4_EMPTY_PACKET_EN	RW	0x0	Enable EP4 empty packet.
1	EP4_IFIFO_CLR	W	0x0	Write 1 to this bit to clear EP4 IN FIFO.
0	EP4_DATA_READY	RW	0x0	The bit is set by MCU when MCU has sent all data to EP4 IN FIFO. The bit is set by USB device controller when IN FIFO has max-packet-size data to transfer. After USB sends all the EP4 data, it will clear this bit. MCU shall poll this bit before sending data to EP4 FIFO, and cannot write data until this bit is cleared.

8.4.4.4.25 EP4_AHBM_RADDR

Name: AHB Master EP4 Reads Start Address Register

Description: This register enables the AHB master EP4 to read the start address.

Base Address: 0x0400A900

Offset: 0x64

Reset Value: 0x00000000

Table 8-193 AHB Master EP4 Reads Start Address Register

Bits	Field Name	RW	Reset	Description
31:0	EP4_RD_SADDR	RW	0x0	AHB master EP4 reads start address.

8.4.4.4.26 EP4_AHBM_CTRL

Name: AHB Master EP4 Reads Control Register

Description: This register controls AHB master EP4.

Base Address: 0x0400A900

Offset: 0x68

Reset Value: 0x0003FF00

Table 8-194 AHB Master EP4 Reads Control Register

Bits	Field Name	RW	Reset	Description
31:18	RSVD	R		Reserved bits
17:8	EP4_AHBM_BURST_SIZE	RW	0x3FF	AHB master EP4 burst length is 32 bytes to 1023 bytes; default: 1023 bytes.
7:1	RSVD	R		Reserved bits
0	EP3_AHBM_EN	RW	0x0	Enable AHB master EP4.

8.4.4.4.27 EP4_XFER_LEN

Name: USB Total EP4 Transfer Length Register

Description: This register sets total transfer length of USB EP4.

Base Address: 0x0400A900

Offset: 0x6C

Reset Value: 0x00000000

Table 8-195 USB Total EP4 Transfer Length Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	EP4_XFER_LEN	RW	0x0	USB total EP4 transfer length. MCU should set this field for AHB transfer.

8.4.4.4.28 EP4_FIFO_WEN

Name: USB EP4 FIFO Writing Byte Enable Register

Description: This register enables writing byte for USB EP4 FIFO.

Base Address: 0x0400A900

Offset: 0x80

Reset Value: 0x00000000

Table 8-196 USB EP4 FIFO Writing Byte Enable Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	EP4_FIFO_WEN	RW	0x1	This register enables writing byte for USB EP4 FIFO

8.4.4.4.29 EP4_FIFO_ADDR

Name: USB EP4 FIFO Address Register

Description: This register sends data to USB EP4 FIFO.

Base Address: 0x0400A900

Offset: 0x84

Reset Value: 0x00000000

Table 8-197 USB EP4 FIFO Address Register

Bits	Field Name	RW	Reset	Description
31:0	USB_EP4_FIFO	W	0x0	USB EP4 FIFO address

8.4.4.4.30 EP5_CTRL

Name: USB EP5 Control Register

Description: This register controls USB EP5.

Base Address: 0x0400A900

Offset: 0x70

Reset Value: 0x00000000

Table 8-198 USB EP5 Control Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	EP5_RX_CNT_NO_OVERWIRTE	RW	0x0	0: EP5_RX_DATA_SUM can be overwritten. 1: EP5_RX_DATA_SUM cannot be overwritten until it has been read.
1	EP5_IFIFO_CLR	W	0x0	Write 1 to this bit to clear EP5 OUT FIFO.
0	EP5_AHBM_EN	RW	0x0	Enable AHB master EP5.

8.4.4.4.31 EP5_AHBM_RADDR

Name: AHB Master EP5 Writing Start Address Register

Description: The register enables the AHB master EP5 to write the start address.

Base Address: 0x0400A900

Offset: 0x74

Reset Value: 0x00000000

Table 8-199 AHB Master EP5 Writing Start Address Register

Bits	Field Name	RW	Reset	Description
31:0	EP5_WR_SADDR	RW	0x0	AHB master EP5 writing start address.

8.4.4.4.32 EP5_XFER_LEN

Name: USB Total EP5 Transfer Length Register

Description: This register sets total transfer length OF USB EP5.

Base Address: 0x0400A900

Offset: 0x78

Reset Value: 0x00000000

Table 8-200 USB Total EP5 Transfer Length Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	EP5_XFER_LEN	RW	0x0	USB total EP5 transfer length. MCU should set this field for AHB transfer.

8.4.4.4.33 EP5_TIMER

Name: EP5 Isochronous Data Out Timer Register

Description: This register indicates the OUT timer value of isochronous data transfer of EP5.

Base Address: 0x0400A900

Offset: 0x7C

Reset Value: 0x00000001

Table 8-201 EP5 Isochronous Data Out Timer Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	USB_EP5_TIMER	RW	0x1	EP5 isochronous out timer value is 1 to 1000 ms. If this field is set to 0, EP5 will never time out.

8.4.4.4.34 EP5_FIFO_ADDR

Name: USB EP5 FIFO Address Register

Description: This register reads USB EP5 FIFO data.

Base Address: 0x0400A900

Offset: 0x88

Reset Value: 0x00000000

Table 8-202 USB EP5 FIFO Address Register

Bits	Field Name	RW	Reset	Description
31:0	USB_EP5_FIFO	R	0x0	USB EP5 FIFO address

8.4.4.4.35 EP5_RX_CNT

Name: USB Endpoint5 Received Data Sum Register

Description: This register gets the received data sum in USB Endpoint5.

Base Address: 0x0400A900

Offset: 0x8C

Reset Value: 0x00000000

Table 8-203 USB Endpoint5 Received Data Sum Register

Bits	Field Name	RW	Reset	Description
31:0	EP5_RX_DATA_SUM	R	0x0	USB received data sum in the EP5 transfer.

8.4.4.4.36 USB_DEBUG

Name: USB Debug Register

Description: This register is used for USB debugging.

Base Address: 0x0400A900

Offset: 0x90

Reset Value: 0x00000150

Table 8-204 USB Debug Register

Bits	Field Name	RW	Reset	Description
31:20	RSVD	R		Reserved bits
19	EP5_OUT_READY_FLAG	WC	0x0	Ep5 output ready flag; write 1 to clear it
18	EP4_DAT_SENT_FLAG	WC	0x0	Ep4 data sent flag; write 1 to clear it
17	EP5_WR_ERR	WC	0x0	Ep5 write err flag; write 1 to clear it
16	EP4_EMTY_PACKET	WC	0x0	Ep4 empty packet flag; write 1 to clear it
15:9	RSVD	R	0x0	Reserved bits
8	EP5_FIFO_EMPTY	R	0x1	Ep5 FIFO empty flag
7	EP4_FIFO_FULL	R	0x0	Ep4 FIFO full flag
6	EP4_FIFO_EMPTY	R	0x1	Ep4 FIFO empty flag
5	EP3_FIFO_FULL	R	0x0	Ep3 FIFO full flag
4	EP3_FIFO_EMPTY	R	0x1	Ep3 FIFO empty flag
3:0	USB_PROBE	R	0x0	State output based on probe_sel[2:0]

8.4.4.4.37 USB_TRX_LDO

Name: USB LDO Control Register

Description: This register controls USB LDO.

Base Address: 0x4000A884

Offset: 0x00

Reset Value: 0x00000001

Table 8-205 USB_TRX_LDO Register

Bits	Field Name	RW	Reset	Description
31:11	RSVD	R		Reserved bits
10:8	USB_RG_LDO3V3_VSEL	RW	0x0	USB LDO trimming is used to adjust LDO3V3 voltage.
7:5	RSVD	R		Reserved bits
6:4	USB_RG_LDO3V3_BIAS_SEL	RW	0	USB LDO trimming is used to adjust D+/D- rise and fall time.
3:1	RSVD	R		Reserved bits
0	USB_RG_LDO3V3_LP_EN	RW	0	1: Enable USB LDO low power mode. 0: Disable USB LDO low power mode.

8.4.4.4.38 USB_LP_CTRL

Name: USB Low Power Control Register

Description: This register controls USB low power mode.

Base Address: 0x4000E140

Offset: 0x00

Reset Value: 0x00000000

Table 8-206 USB_LP_CTRL Register

Bits	Field Name	RW	Reset	Description
31:9	RSVD	R		Reserved bits
8	USB_CLK_FORCE_OFF	RW	0x0	0: Follow usb_suspend_clk_off. 1: Force usb_clk off.
7:5	RSVD	R		Reserved bits
4	USB_PMU_LP_EN	RW	0	0: Disable USB PMU low power mode. 1: Enable USB PMU low power mode.
3:1	RSVD	R		Reserved bits
0	USB_SUSPEND_CLK_OFF	RW	0	Turn off USB clock during USB suspend.

8.4.4.4.39 USB_XCVR_CTRL

Name: USB Transceiver Control Register

Description: This register controls USB transceiver.

Base Address: 0x4000E140

Offset: 0x04

Reset Value: 0x0000000F

Table 8-207 USB_XCRV_CTRL Register

Bits	Field Name	RW	Reset	Description
31:9	RSVD	R		Reserved bits
10:8	USB_RG_XCVR_RTRIMP	RW	0	USB D+ trimming is used to adjust D+ impedance.
6:4	USB_RG_XCVR_RTRIMN	RW	0	USB D- trimming is used to adjust D- impedance.
3	USB_RG_XCVR_BIAS_EN	RW	1	Reserved bits
2	USB_RG_XCVR_SUSPEND	RW	1	USB suspend control 0: Disable 1: Enable
1	USB_RG_XCVR_SPEED	RW	1	1: Full speed 0: Slow speed
0	USB_RG_XCVR_SECMP_PD	RW	1	USB single end comparator power down control 0: Power up 1: Power down

8.4.4.4.40 USB_SW_RST

Name: USB Module All Reset Control Register

Description: This register controls USB reset.

Base Address: 0x4000E140

Offset: 0x0C

Reset Value: 0x00000000

Table 8-208 USB_SW_RST Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	USB_SOFTWARE_RESET	W	0	Write 1 to reset USB. No need to manually write it back to 0.

8.4.4.5 Electrical Specifications

Table 8-209 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
IUSB,ACTIVE,VBUS	Current from VBUS supply, USB active		2.7		mA
IUSB,SUSPEND,VBUS	Current from VBUS supply, USB suspended		360		µA
IUSB,DISABLED,VBUS	Current from VBUS supply, USB disabled (low power mode)	0.8	1.1	1.3	µA
RUSB,PU,ACTIVE	Value of pull-up resistance on D+, bus active	1425	2600	3090	Ω

Symbol	Description	Min.	Typ.	Max.	Unit
RUSB,PU,IDLE	Value of pull-up resistance on D+, bus idle	900	1500	1575	Ω
fUSB,CLK	Frequency of local clock, USB active		48		MHz

8.4.5 ISO/IEC 7816-3 Master

8.4.5.1 Introduction

GR5526 SoC has one instance of SIM interface block that implements ISO/IEC 7816 standard. ISO/IEC 7816 is an international standard related to electronic identification cards with contacts, especially smart cards, managed jointly by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).

8.4.5.2 Main Features

- UICC power save (clock stop)
- UICC cold and warm reset
- Guard time handling
- Wait time evaluation
- Low power design

8.4.5.3 Functional Description

The SIM interface module provides an interface to an external Universal Integrated Circuit Card (UICC) containing the Universal Subscriber Identity Module (USIM) application. The corresponding software module implements the SIM terminal transport, Card Application Toolkit (CAT), and application layer. It has a software interface to access all required USIM functions.

8.4.5.3.1 Block Diagram

The block diagram of SIM Interface is illustrated in Figure 8-41. A main clock gate disables the module clock during any module inactivity for a minimal dynamic power consumption. The SIM clock is derived from the module clock and base for the Elementary Time Unit (ETU) generation. SIM clock and ETU are tracked by counters.

A shift register in the I/O module de-/serializes the bit stream and calculates/checks the parity. Data bytes are served by/to a Byte Direct Memory Access (DMA). The overall transmission is managed by the control unit. Events are signaled via interrupts. The UICC presence is monitored by the presence observer module.

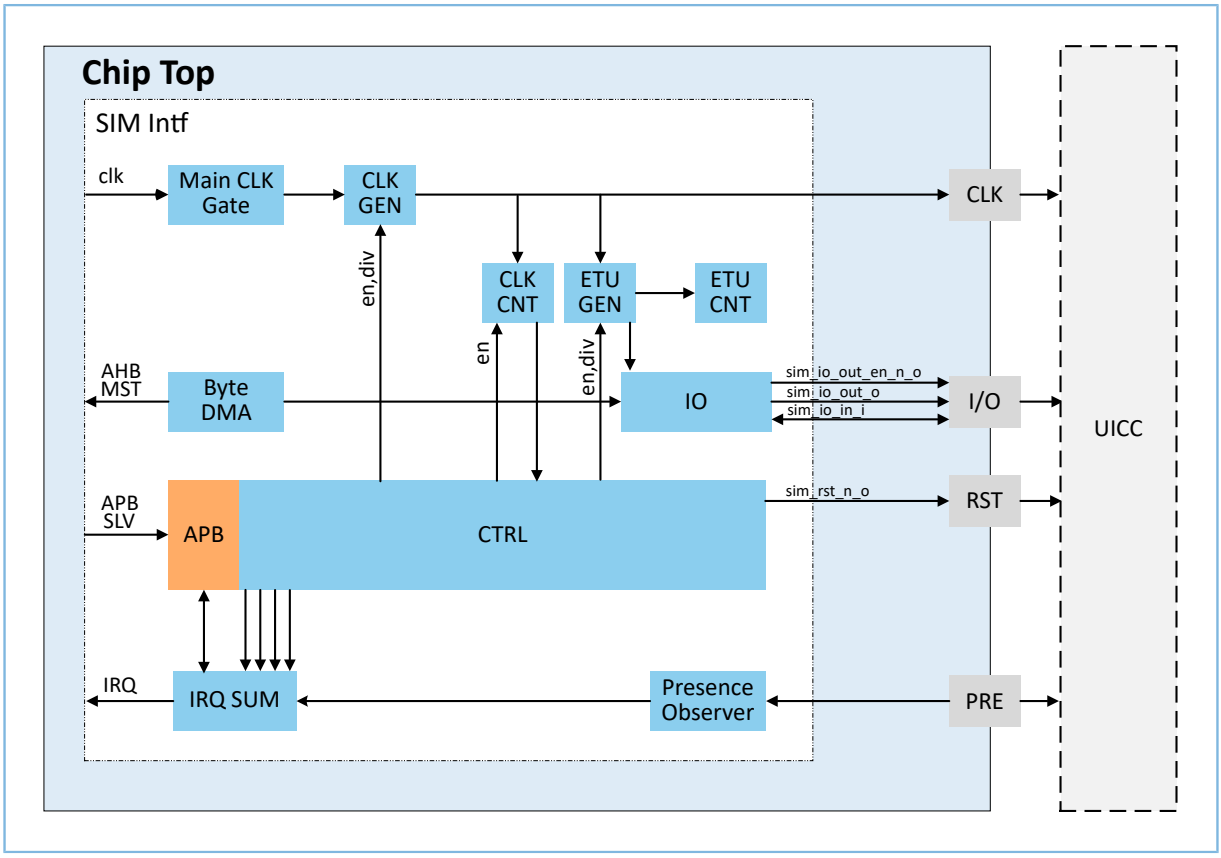

Figure 8-41 SIM Interface block diagram

8.4.5.3.2 Hardware Behavior

The SIM interface hardware module supports the physical, data link, and transport layer defined in *ETSI TS 131 101*.

8.4.5.3.2.1 Physical Layer

The SIM interface hardware module interfaces the UICC with the following ports:

- Clock (CLK)
- Reset (RST)
- Input/Output (I/O)
- UICC presence (PRE)

The detailed electrical required characteristic of the ports can be taken from *ETSI TS 131 101*.

8.4.5.3.2.1.1 CLK

The SIM clock is derived from dividing the module clock by CLK_CFG.CLK_DIV.

$$f = \frac{f_{mod}}{clk_{div} + 1} \tag{14}$$

The SIM clock frequency shall be in the range of 1 to 5 MHz, 4 MHz during UICC initialization and 5 MHz during normal operation. The clock division has to be chosen accordingly.

8.4.5.3.2.1.2 RST

The UICC will be reset by the RST pin of the SIM interface module, and the RST is controlled by the register CTRL.ACTION.

8.4.5.3.2.1.3 I/O

The bidirectional I/O pad on chip top level is implemented as open drain with a recommended pull up to VCC of 20 kΩ. If supported by the UICC, the I/O pad can use low-impedance drivers to save power.

8.4.5.3.2.1.4 PRE

UICC holder often serve a card presence contact. Any card presence change will trigger an interrupt (see STAT.PRESENCE_STAT and STAT.IRQ_PRESENCE in the STAT register).

8.4.5.3.2.2 Power States

The SIM interface module is responsible for executing the cold reset, warm reset, clock stop and the power down sequence. Software triggers power state changes by writing register field CTRL. ACTION.

8.4.5.3.2.2.1 Activation on Cold Reset

The activation sequence and cold reset sequence is shown in Figure 8-42. At first VCC is ramped up. With T_a the clock starts while the reset is still asserted. On T_b the reset is released and the UICC has to start with the ATR within t_c.

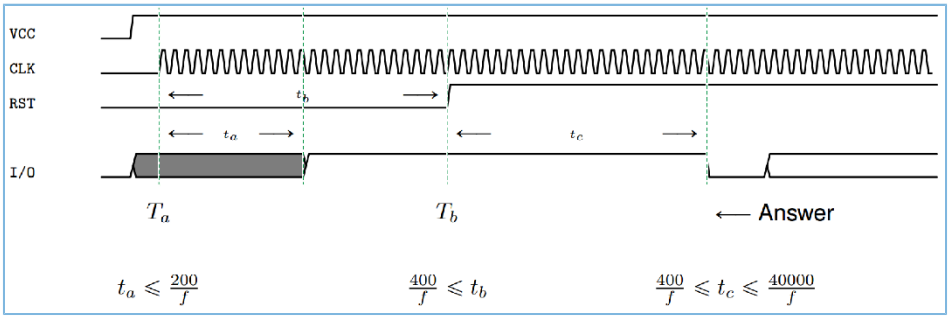

Figure 8-42 Activation on cold reset

The I/O pin is not actively driven by the SIM interface during the entire cold reset procedure. The high level is ensured by the pull-up resistor. Therefore t_a is not implemented in the SIM interface.

t_b is hard-coded to 400 clock cycles.

The minimum t_c is not checked by the SIM interface module.

The maximum t_c is defined by register fields CLK_CFG.ETU_DIV and TIMES_CFG.WAIT_TIME with respect to the following equation.

$$t_c = \frac{(\mathrm{etu_{div}} + 1) \times (\mathrm{waittime} + 1)}{f_{\mathrm{mod}}} \tag{15}$$

The reset values are chosen according the specification to receive the ATR without any modifications. CLK_CFG.ETU_DIV is 371 and TIMES.WAITTIME is 107 (see the equation below).

$$waittime = \left\lceil \frac{40000}{\mathrm{etu_{div}} + 1} \right\rceil = \left\lceil 106.52 \right\rceil = 107 \tag{16}$$

If maximum t_c is not met by the UICC, interrupt STAT.IRQ_RX_ERR is triggered.

8.4.5.3.2.2.2 Warm Reset

The warm reset sequence is shown in Figure 8-43.

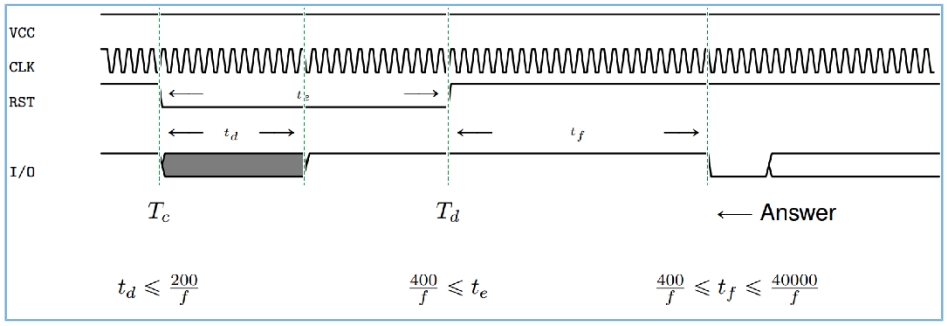

Figure 8-43 Warm reset

The warm reset is implemented identically to the cold reset, with $T_c = T_a$, $T_d = T_b$, $t_d = t_a$, $t_e = t_b$ and $t_f = t_c$.

Due to the fact that the registers CLK_CFG and TIMES_CFG may have changed after the ETU evaluation, the reset values have to be restored by firmware before triggering the warm reset.

8.4.5.3.2.2.3 Clock Stop

The clock stop sequence is shown in Figure 8-44.

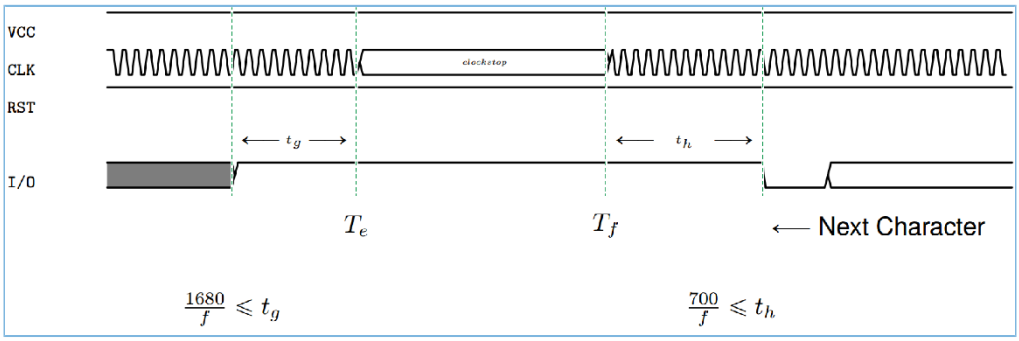

Figure 8-44 Clock stop

The times t_g and t_h are ensured by the SIM interface hardware module, with $t_g = \frac{1680}{f}$ and $t_h = \frac{700}{f}$.

8.4.5.3.2.2.4 Deactivation

The UICC deactivation sequence is shown in Figure 8-45.

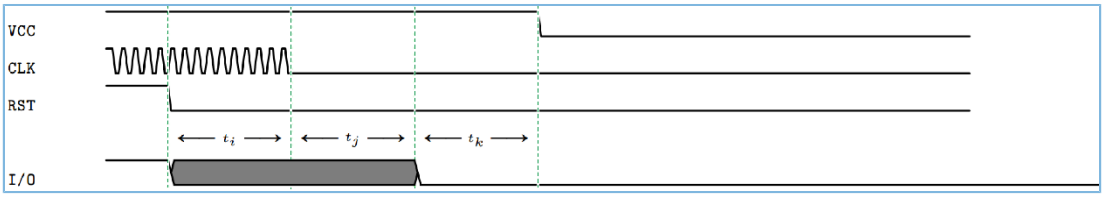

Figure 8-45 Deactivation

The times t_i and t_j are ensured by the SIM interface hardware module, with $t_i = \frac{100}{f}$ and $t_j = \frac{100}{f_{mod}}$. The timing t_k depends on the VCC source.

8.4.5.3.2.3 Data Link Layer

8.4.5.3.2.3.1 Elementary Time Unit (ETU)

The nominal duration of one moment on the electrical circuit I/O is named Elementary Time Unit (ETU).

$$\frac{F}{D} * \frac{1}{f} \tag{17}$$

The ETU shall be equal to F/D clock cycles on the electrical circuit CLK where F and D are the transmission parameters. F is the clock rate conversion integer and D is the baud rate adjustment integer. F, D and f are negotiated during the initialization and setup by firmware. Possible values for F and D are shown in Table 8-210.

Table 8-210 Values for F and D

F	D	F/D
372	1	372
372	2	186
372	3	124
372	4	93
512	1	512
512	2	256
512	4	128
512	8	64
512	16	32
512	32	16
512	64	8

f is configured as described in “Section 8.4.5.3.2.1.1 CLK”.

The chosen ETU has to be written to the register field CLK_CFG.ETU_DIV.

$$\mathrm{etu}_{div} = \frac{F}{D} - 1 \tag{18}$$

Fractional ETU values are not supported.

8.4.5.3.2.3.2 Character Frame

The character transmission timing is shown in Figure 8-46. A character consists of:

- Start Bit
- Data Bits
- Parity Bit
- 2 Stop Bits with error signaling

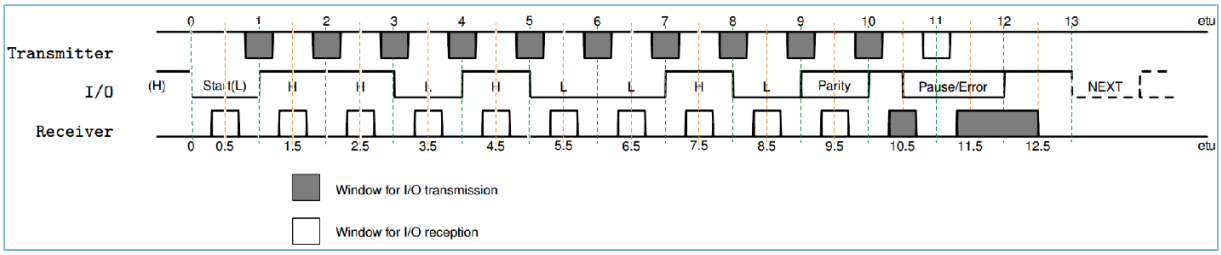

Figure 8-46 Character timing

8.4.5.3.2.3.3 Error Signal and Character Repetition

- RX

  In case of detected parity error during reception, the RX behavior depends on register field STAT.RX_RETRY_MAX. If the number of retries is not exceeded, the I/O pin is pulled low for one clock cycle to trigger a retransmission. Otherwise or if RETRY_MAX = 0, the received byte is discarded and the reception is aborted with a STAT.IRQ_RETRY_ERR interrupt. The internal retry counter is restarted with every character.

  The maximum number of retries during the reception of one byte is displayed in register field STAT.RX_RETRY_MAX and cleared via CTRL.RX_RETYR_MC.

- TX

  In case of an error signaling from the receiver, the TX behavior depends on register field STAT.TX_RETRY_MAX. If the number of retries is not exceeded, the character is retransmitted. Otherwise or if RETRY_MAX = 0, the transmission is aborted with a STAT.IRQ_RETRY_ERR interrupt. The internal retry counter is restarted with every character.

  The maximum number of retries during the transmission of one byte is displayed in register field STAT.TX_RETRY_MAX and cleared via CTRL.TX_RETYR_MC.

8.4.5.3.2.3.4 Guard Time and Wait Time

The guard time specifies the minimum time between the leading edge of two consecutive characters, and the time has to be taken into account by the transmitter. The wait time specifies the maximum time between the leading edge of two consecutive characters, and the time has to be monitored by the receiver before issuing a timeout.

The guard time is adjusted by register field TIMES_CFG.GUARD_TIME and respected before transmitting a character. This includes the scenarios when the SIM interface has received a character and starts a transmission afterwards. The minimal guard time is 12. Register field TIMES_CFG.GUARD_TIME values smaller than that are ignored.

The wait time is adjusted by register field TIMES_CFG.WAIT_TIME and respected before aborting a character reception.

8.4.5.3.2.4 Answer to Reset

After a cold or warm reset, the UICC answers with the Answer to Reset (ATR). The module receives the ATR and decodes the first byte TS. TS marks the used encoding. If enabled in register field DATA_CFG.DETECT_CODING, the coding is set accordingly in DATA_CFG.CODING. All subsequent bytes are decoded with ISO/IEC 7816-3.

The class selection procedure and the Protocol and Parameter Select (PPS) (as a response to the ATR) procedure is controlled by firmware. After the PPS procedure, the firmware is responsible for configuring the transport protocol, clock frequency, I/O mode, and ETU.

8.4.5.3.2.5 Receive and Transmit

The SIM interface hardware module writes all received data bytes via a Byte DMA to a memory. Transmitted data bytes are read via the same Byte DMA.

8.4.5.3.2.5.1 Byte DMA

The Byte DMA accesses memory addresses word-wise and interacts with the I/O module on a byte base. In transmitting process, one entire word is read and served byte-wise. Irrelevant bytes are simply ignored. In receiving process, the word is cleared at the beginning and filled byte-wise. After receiving the fourth byte or after receiving the last byte, the entire word is written to the memory.

8.4.5.3.2.5.2 Address Ranges

Receive and transmit buffers share the same start address (register START_ADDR), which has to be aligned to 4.

The internal word address counting is limited to 18 bits. Therefore, the maximum receive and transmit buffer sizes are limited to $2^{(18+2)}$ Bytes (1 MB). The buffers shall not cross a 1 MB address boundary.

8.4.5.3.2.5.3 Receive Buffer

The receive buffer end address is controlled by register field RX_END_ADDR and has to be calculated according to the following equation for an expected number of bytes N_{RX}.

$$rx_{end_{addr}} = strt_{addr} + N_{RX} - 1 \tag{19}$$

A done interrupt will be issued after receiving N_{RX} bytes.

If less bytes are received and the wait time is expired, an rx_err interrupt is triggered. The register address (addr) points to the last received byte, which can be used to calculate the number of received bytes n_{rx}.

$$n_{rx} = addr - strt_{addr} + 1 \tag{20}$$

If more bytes are served by the UICC than expected, an rx_err interrupt is triggered, but the incoming bytes are ignored and addr stays untouched.

8.4.5.3.2.5.4 Transmit Buffer

The transmit buffer end address is controlled by register field TX_END_ADDR and has to be calculated according to the following equation for an expected number of bytes N_{TX}.

$$tx_{end_{addr}} = strt_{addr} + N_{TX} - 1 \quad (21)$$

8.4.5.3.2.5.5 DMA Error

A DMA error interrupt dma_err is triggered under any of the following conditions:

- Invalid address
- Bus slave responses with an error
- Word could not be read within one ETU
- Word could not be written within one character

8.4.5.3.2.6 Action State

By default, the SIM interface hardware module does nothing, unless the firmware requests an action by writing register field CTRL.ACTION.

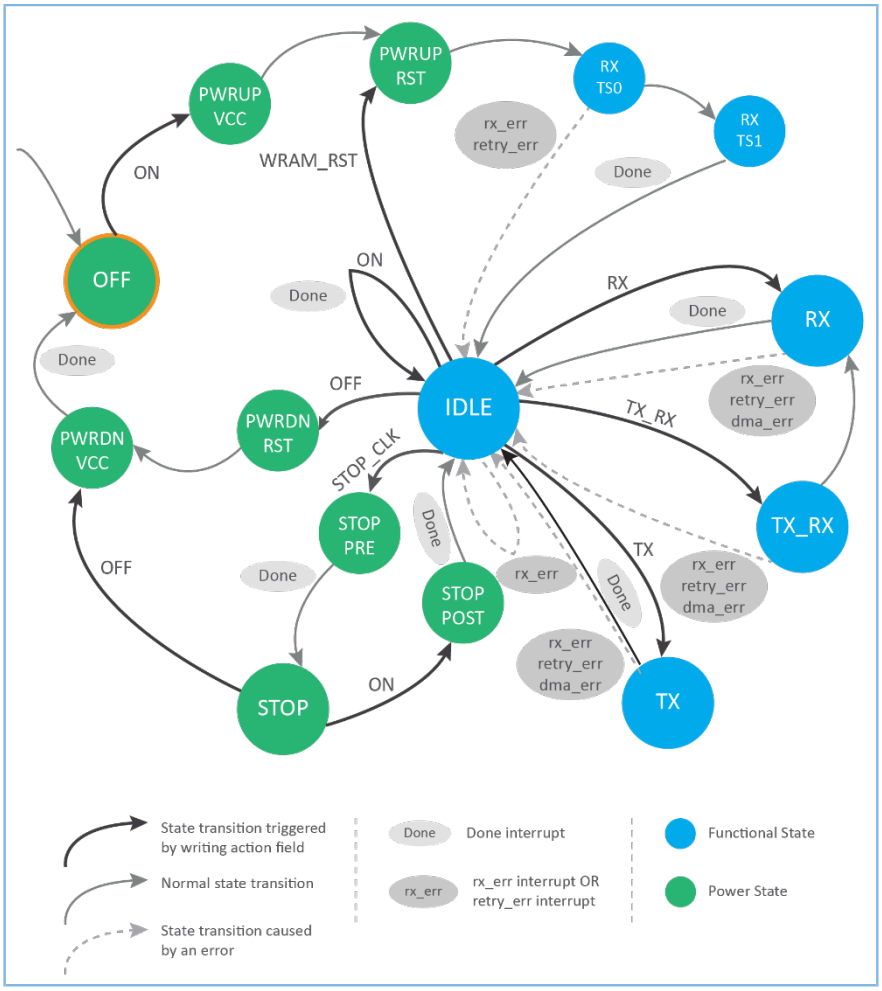

Figure 8-47 Main SIM Interface states

The main state machine is shown in Figure 8-47, with the following major states:

- OFF: UICC is powered off.
- IDLE: UICC is powered on and ready to communicate.
- RX: Receive characters from UICC and store them in the receive buffer.
- TX: Read characters from the transmit buffer and transmit them to UICC.
- TX_RX: Read characters from the transmit buffer and transmit them to UICC, continue with RX.
- STOP: UICC is powered on but clock is stopped.

The main state machine operation bases on the following rules:

- The reset state is OFF.
- The action NONE (0) does nothing.
- The states OFF, IDLE, and STOP are inactive states with BUSY = 0.
- All other states are active states with BUSY = 1.

- The busy flag is visible in register field STAT.BUSY.
- It is only allowed to request the state transitions in Figure 8-47.
- Any other action will trigger a state_err interrupt, but not abort the ongoing action.
- On entering of an inactive state a done interrupt is triggered (if no error occurs).
- In case of an rx_err, retry_err or dma_err error, the corresponding interrupt is issued, and the idle state is entered.
- Requesting ON during IDLE does nothing except triggering a done interrupt.

8.4.5.3.2.7 Transport Layer

The module supports the automatic transmission and reception of multiple bytes. With the in-built DMA the TX data is take from or the RX data is written directly into the memory. For TX, the firmware writes the payload into the memory and configures the module with payload start address and payload length. The module transfers the payload automatically and signalizes the end of the transmission with an interrupt. For RX, the firmware configures the module with expected payload length and payload address. The module writes the received payload at the configured memory address. The number of received bytes is stored inside the module and interrupts for both the beginning of transmission and end of transmission are generated.

ISO/IEC 7816-3 defines three types of data transport modes:

- Plain mode: In plain mode, the module transmits or receives multiple bytes controlled by software. Thus, the module is explicitly set in TX, RX or TX followed by RX mode.
- T = 0 mode: T = 0 mode is not implemented in hardware on purpose. SIM implementations vary in details. Therefore, the implementation is done in firmware using plain mode.
- T = 1 mode: T = 1 mode is not implemented in hardware on purpose. SIM implementations vary in details. Therefore, the implementation is done in firmware using plain mode.

8.4.5.4 Registers

8.4.5.4.1 CTRL

- **Name:** Control Register
- **Description:** This register is used to control the interrupts and TX/RX retries of the SIM interface module. This register is only for write purposes.
- **Base Address:** 0x4000F200
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 8-211 Control Register

Bits	Field Name	RW	Reset	Description
31	IRQ_TEST_SET	W	0x0	Interrupt Test Set.

Bits	Field Name	RW	Reset	Description
				This register sets test interrupt. **Value:** • 0x0: No effect • 0x1: Set
30	IRQ_TEST_CLR	W	0x0	Interrupt Test Clear. This register clears test interrupt (has higher priority than 'IRQ_TEST_SET') **Value:** • 0x0: No effect • 0x1: Clear
29:26	RSVD	W		Reserved bits
25	IRQ_PRESENCE_CLR	W	0x0	Interrupt source presence clear. This register clears interrupt source 'STAT.IRQ_PRESENCE'. **Value:** • 0x0: No effect • 0x1: Clear
24	IRQ_STAT_EC	W	0x0	Interrupt source state_err clear. This register clears interrupt source 'STAT.IRQ_STAT_ERR'. **Value:** • 0x0: No effect • 0x1: Clear
23	IRQ_DMA_EC	W	0x0	Interrupt source dma_err clear. This register clears interrupt source 'STAT.IRQ_DMA_ERR'. **Value:** • 0x0: No effect • 0x1: Clear
22	IRQ_RETYR_EC	W	0x0	Interrupt source retry_err clear. This register clears interrupt source 'STAT.IRQ_RETYR_ERR'. **Value:** • 0x0: No effect • 0x1: Clear
21	IRQ_RX_EC	W	0x0	Interrupt source rx_err clear. This register clears interrupt source 'STAT.IRQ_RX_ERR'. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0: No effect • 0x1: Clear
20	IRQ_DONE_CLR	W	0x0	Interrupt source done clear. This register clears interrupt source 'STAT.IRQ_DONE'. **Value:** • 0x0: No effect • 0x1: Clear
19:13	RSVD	W		Reserved bits
12	TX_RETYR_MC	W	0x0	Transmit Retries Maximum Clear. This register clears 'STAT.TX_RETRY_MAX'. **Value:** • 0x0: No effect • 0x1: Clear
11:9	RSVD	W		Reserved bits
8	RX_RETYR_MC	W	0x0	Receive Retries Maximum Clear. This register clears 'STAT.RX_RETRY_MAX'. **Value:** • 0x0: No effect • 0x1: Clear
7:3	RSVD	W		Reserved bits
2:0	ACTION	W	0x0	**Value:** • 0x0: Do Nothing. • 0x1: Switch off. • 0x2: Stop the clock. • 0x3: Switch on and receive ATR. Re-enable clock if clock is stopped. • 0x4: Trigger warm reset and receive ATR. • 0x5: Receive. • 0x6: Transmit. • 0x7: Transmit, followed by RX.

8.4.5.4.2 STAT

- **Name:** Status Register
- **Description:** This register records interrupts, I/O, and power status.

- **Base Address:** 0x4000F200
- **Offset:** 0x04
- **Reset Value:** 0x00020000

Table 8-212 Status Register

Bits	Field Name	RW	Reset	Description
31	RSVD	R		Reserved bits
30	IRQ_TEST	R	0x0	Interrupt Test. Test interrupt for connection check. **Value:** • 0x0: No interrupt • 0x1: Active
29:26	RSVD	R		Reserved bits
25	IRQ_PRESENCE	R	0x0	Interrupt Presence. SIM card presence changed: Inserted or removed. **Value:** • 0x0: No interrupt • 0x1: Active
24	IRQ_STAT_ERR	R	0x0	Interrupt State Error. Action requested while busy or unsupported transition. **Value:** • 0x0: No interrupt • 0x1: Active
23	IRQ_DMA_ERR	R	0x0	Interrupt DMA Error. DMA read/write operation could not be issued. **Value:** • 0x0: No interrupt • 0x1: Active
22	IRQ_RETRY_ERR	R	0x0	Interrupt Retry Error. Maximum number of retries exceeded. **Value:** • 0x0: No interrupt • 0x1: Active
21	IRQ_RX_ERR	R	0x0	Interrupt RX Error. No or incomplete or unexpected data.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0: No interrupt • 0x1: Active
20	IRQ_DONE	R	0x0	Interrupt Done. Requested operation has been completed. **Value:** • 0x0: No interrupt • 0x1: Active
19:18	RSVD	R		Reserved bits
17	PRESENCE_STAT	R	0x1	Status of presence I/O. **Value:** • 0x0: Absent • 0x1: Presence
16	BUSY	R	0x0	Status of SIM interface. **Value:** • 0x0: Idle • 0x1: Busy
15	RSVD	R		Reserved bits
14:12	TX_RETRY_MAX	R	0x0	Maximum Transmit Retries. Maximum number of seen transmit retries after error signaling by SIM.
11	RSVD	R		Reserved bits
10:8	RX_RETRY_MAX	R	0x0	Maximum Receive Retries. Maximum number of seen receive retries after parity error.
7	RSVD	R		Reserved bits
6:4	IO_STAT	R	0x0	IO States. **Value:** • 0x0: IO pin is driven low. • 0x1: IO pin is high impedance and driven high by the pull-up. • 0x4: Wait for start bit. • 0x5: Receive. • 0x6: Transmit.

Bits	Field Name	RW	Reset	Description
				• 0x7: Wait until next character to ensure configured guard time.
3:0	PWR_STAT	R	0x0	Power States. **Value:** • 0x0: SIM is unpowered. • 0x1: Power up SIM. RST asserted (low). Clock stopped. I/O is tristate. • 0x2: Power up SIM. RST asserted (low). Clock is running. I/O is high. • 0x3: Power up SIM. RST asserted (low). Clock is running. I/O is tristate. • 0x4: Power up SIM. RST asserted (low). Clock stopped. I/O is low. • 0x5: Preparing clock stop. • 0x6: Clock stopped. • 0x7: Exiting clock stop • 0x8: SIM is idle; no communication is ongoing. • 0x9: RX TS0 Character. • 0xA: RX TS1 Character. • 0XB: Receive. • 0xC: Transmit. • 0xD: Transmit and Receive.

8.4.5.4.3 CLK_CFG

- **Name:** Clock Configuration Register
- **Description:** This register configures clock.
- **Base Address:** 0x4000F200
- **Offset:** 0x08
- **Reset Value:** 0x002F0173

Table 8-213 Clock Configuration Register

Bits	Field Name	RW	Reset	Description
31	CLK_STOP_SEL	RW	0x0	Clock Stop Select. Value of the clock output during stopped clock. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0: Low • 0x1: High
30:24	RSVD	R		Reserved bits
23:16	CLK_DIV	RW	0x2F	Clock Division. Divide system clock by this value + 1.
15:10	RSVD	R		Reserved bits
9:0	ETU_DIV	RW	0x173	Divide SIM clock by this value + 1 to define ETU length. The reset value is the one, needed for the ATR.

8.4.5.4.4 TIMES_CFG

- **Name:** Times Configuration Register
- **Description:** This register configures the timing.
- **Base Address:** 0x4000F200
- **Offset:** 0x10
- **Reset Value:** 0x006B0000

Table 8-214 Times Configuration Register

Bits	Field Name	RW	Reset	Description
31:30	RSVD	R		Reserved bits
29:12	WAIT_TIME	RW	0x6B	Wait time in [ETU]. Maximum card response time (leading edge to leading edge).
11:10	RSVD	R		Reserved bits
9:0	GUARD_TIME	RW	0x0	Guard time in [ETU]. Time between the leading edges of two consecutive characters.

8.4.5.4.5 DATA_CFG

- **Name:** Data Configuration Register
- **Description:** This register configures the communication.
- **Base Address:** 0x4000F200
- **Offset:** 0x14
- **Reset Value:** 0x00000002

Table 8-215 Data Configuration Register

Bits	Field Name	RW	Reset	Description
31:7	RSVD	R		Reserved bits
6:4	RETRY_LIMIT	RW	0x0	Retries Limit. Maximum number of issued retries before giving up.
3:2	RSVD	R		Reserved bits
1	DETECT_CODING	RW	0x1	Detect Coding Convention. Automatically detect coding convention during ATR reception. **Value:** • 0x0: Disable • 0x1: Enable
0	CODING	RW	0x0	Coding Convention. **Value:** • 0x0: Default. High = 1, LSB first. • 0x1: Inverse. High = 0, MSB first.

8.4.5.4.6 ADDR

- **Name:** Address Register
- **Description:** This register records current address relative to base_addr.
- **Base Address:** 0x4000F200
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 8-216 Address Register

Bits	Field Name	RW	Reset	Description
31:20	RSVD	R		Reserved bits
19:2	ADDR	R	0x0	Address. Current address relative to base_addr.
1:0	ADDR_FRAC	R	0x0	Address Fraction. Byte selection.

8.4.5.4.7 START_ADDR

- **Name:** Start Address Register

- **Description:** This register configures read/write memory address. RX and TX buffers have to be aligned to 4 bytes, and buffer address [31:2] = START_ADDR[31:2]
- **Base Address:** 0x4000F200
- **Offset:** 0x1c
- **Reset Value:** 0x00000000

Table 8-217 Start Address Register

Bits	Field Name	RW	Reset	Description
31:20	BASE_ADDR	RW	0x0	Base Address. Base Address for RX and TX buffers.
19:2	START_ADDR	RW	0x0	Start Address. Start address of RX and TX buffers, relative to base_addr.
1:0	RSVD	R		Reserved bits

8.4.5.4.8 RX_END_ADDR

Name: RX End Register

Description: This register configures RX data end address.

Base Address: 0x4000F200

Offset: 0x20

Reset Value: 0x00000000

Table 8-218 RX End Register

Bits	Field Name	RW	Reset	Description
31:20	RSVD	R		Reserved bits
19:2	RX_END_ADDR	RW	0x0	RX End Address. End address of receive buffer, relative to base_addr.
1:0	RX_END_AF	RW	0x0	RX End Address Fraction. Byte selection.

8.4.5.4.9 TX_END_ADDR

Name: TX End Register

Description: This register configures TX data end address.

Base Address: 0x4000F200

Offset: 0x24

Reset Value: 0x00000000

Table 8-219 TX End Register

Bits	Field Name	RW	Reset	Description
31:20	RSVD	R		Reserved bits
19:2	TX_END_ADDR	R	0x0	TX End Address. End address of transmit buffer, relative to base_addr.
1:0	TX_END_AF	R	0x0	TX End Address Fraction. Byte selection.

8.4.5.5 Electrical Specifications

Table 8-220 Electrical Specifications

Symbol	Description	Min.	Typ.	Max.	Unit
f_{SIMCLK}	UICC CLK frequency	1		5	MHz
VCC	UICC Voltage class supported	1.8		3.08	V

8.5 Display Interfaces

8.5.1 DSPI

8.5.1.1 Introduction

DSPI (Display SPI) block implements a display interface that complies with MIPI DBI Type C standard. It is based on SPI protocol and incorporates protocol extensions that are common in the industry. It is important to note that DSPI module is not intended to be used as a general-purpose SPI Master.

8.5.1.2 Main Features

Features of the DSPI controller include:

- Master operation (in Display applications, the host always serves as the master and only supports write operations.)
- Multiple transmission options: 3W1L, 4W1L, and 4W2L
- Data size options enhanced to ensure [8, 9, 16, 18, 24] bits to support transmission of common pixel formats
- Baud rate prescalers up to fPCLK/2
- Chip selection signal CS management by hardware or software
- Dedicated transmission and reception flags with interrupt capability
- DMA supported
- Bus busy status flag

- 16-byte TX FIFOs with DMA capability

8.5.1.3 Functional Description

DSPI allows synchronous, serial communication between the host and external display module. Application software can manage the communication by polling the status flag or using a DSPI interrupt. The main elements of DSPI and their interactions are shown in the following block diagram.

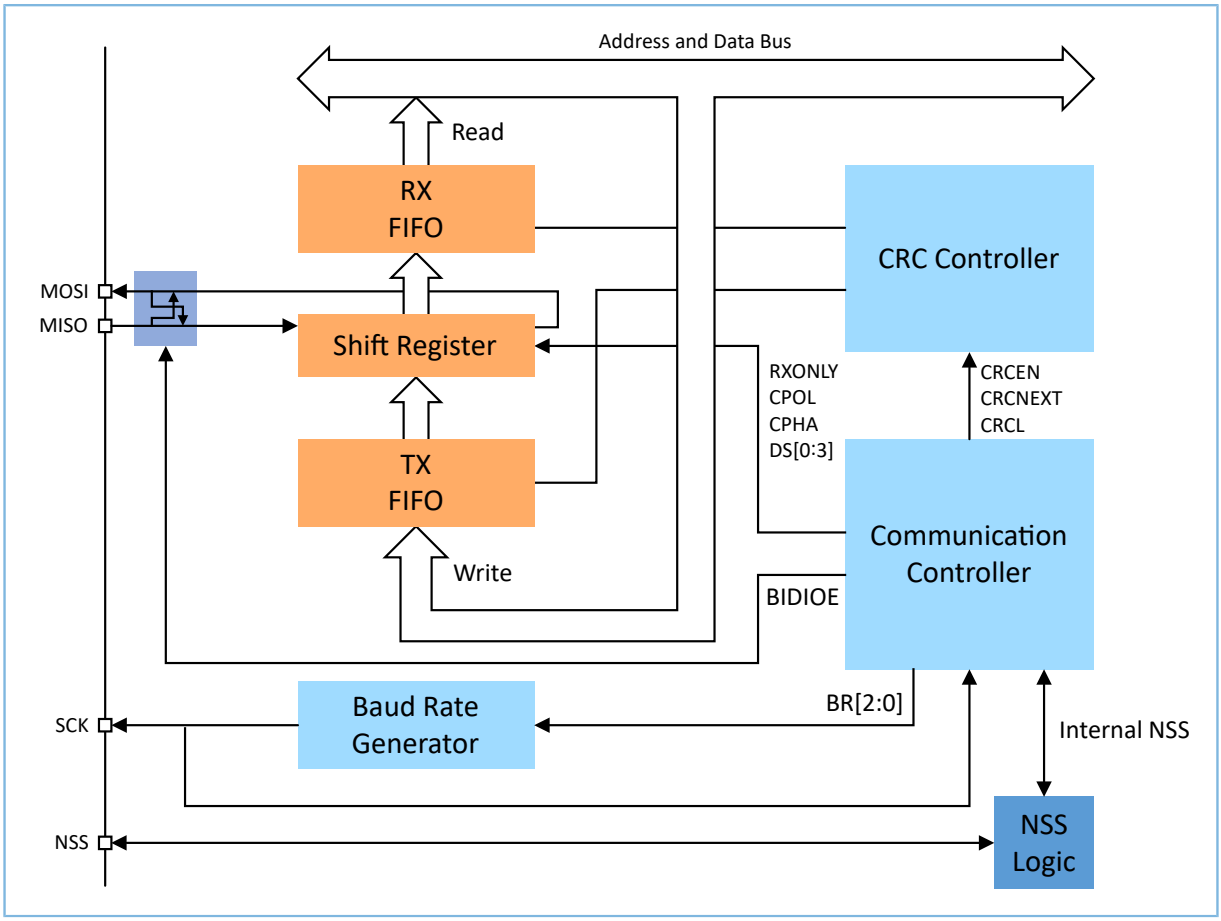

Figure 8-48 DSPI block diagram

8.5.1.3.1 Protocols Introduction

DSPI supports the following three transfer modes for data transmission to display module, which is a serial data write.

- 3W1L (3-Wire 1-Lane) display interface features 3 interface wires & 1 data lane.
- 4W1L (4-Wire 1-Lane) display interface features 4 interface wires & 1 data lane.
- 4W2L (4-Wire 2-Lane) display interface features 4 interface wires & 2 data lanes.

8.5.1.3.1.1 3W1L Display Interface [A0+8-bit]

3W1L interface in [A0+8-bit] mode, illustrated in the figure below, allows putting a total of 9 bits per transaction on the MOSI wire, in which

- First bit ("A0") is the data-versus-command indicator bit.
- Next 8 bits (D7 through D0) are data bits.

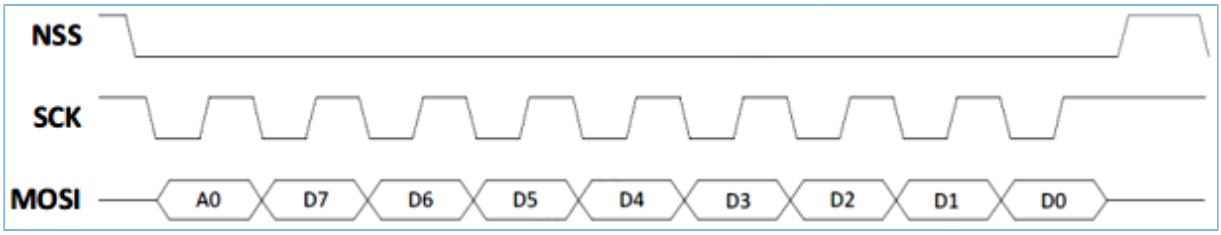

Figure 8-49 3W1L interface in [A0+8-bit] mode

8.5.1.3.1.2 4W1L Display Interface [8-bit]

4W1L interface in [8-bit] mode, illustrated in the figure below, allows putting

- 8 bits per transaction on the MOSI wire, in which all bits (D7 through D0) are data bits.
- Data-versus-command information on the dedicated DCX wire, for the duration of the transfer.

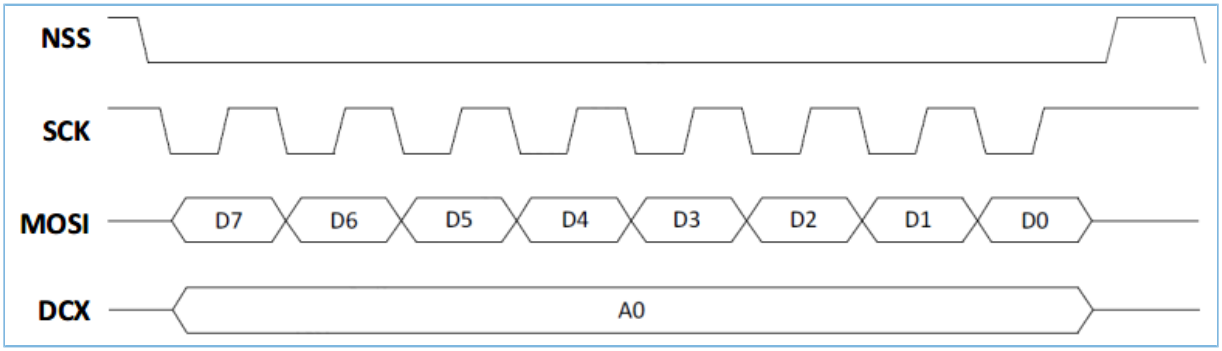

Figure 8-50 4W1L interface in [8-bit] mode

8.5.1.3.1.3 4W2L Display Interface [A0+8-bit]

4W2L interface in [A0+8-bit] mode, illustrated in the figure below, allows putting a total of 18 bits per transaction on the MOSI and DCX wires, in which

- First bit ("A0") on both wires is the data-versus-command indicator bit. This bit is duplicated.
- Next 8 bits (D7 through D0) on MOSI wire are data bits.
- Next 8 bits (D15 through D8) on DCX wire are data bits.

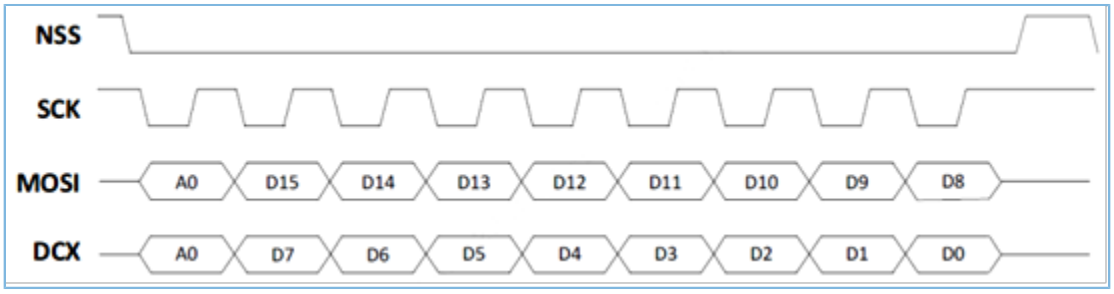

Figure 8-51 4W2L interface in [A0+8-bit] mode

8.5.1.3.2 Choice of DS and SPIMODE for Target Pixel Sizes

There are various choices of DS[4:0] and SPIMODE[1:0] when users desire to send pixels of a particular format. The choice mostly depends on the Display Module capabilities (number of wires and data lanes and pixel formats supported) and the desired level of performance (throughput).

The DMA transfer size, which is a setting outside of the DSPI IP, is also an important parameter, because it determines the granularity of data that is written into DSPI TX FIFO.

Typical choices of DS and SPIMODE for target pixel sizes are shown in Table 8-221. This table applies to 3 common pixel formats: RGB565, RGB666, and RGB888.

Table 8-221 Typical choices of DS and SPIMODE for target pixel sizes

Pixel Format	Pixel Size	Transfers per Pixel	DSPI Transfer Description	DMA Data Size DST_TR_WIDTH[2:0]	Transfer Mode SPIMODE[1:0]	DSPI Data size DS[4:0]
RGB565	16b	2	A0 + 8b on 1 lane	0x0: 8 bits	0x1: 3W1L	0x07: 8 bits
		1	A0 + 16b on 1 lane	0x1: 16 bits	0x1: 3W1L	0x0F: 16 bits
		2	8b on 1 lane	0x0: 8 bits	0x2: 4W1L	0x07: 8 bits
		1	A0 + 8b on 2 lanes	0x1: 16 bits	0x3: 4W2L	0x0F: 16 bits
RGB666	18b	3	A0 + 8b on 1 lane	0x0: 8 bits	0x1: 3W1L	0x07: 8 bits
		2	A0 + 9b on 1 lane	0x1: 16 bits	0x1: 3W1L	0x08: 9 bits
		1	A0 + 18b on 1 lane	0x2: 32 bits	0x1: 3W1L	0x11: 18 bits
		3	8b on 1 lane	0x0: 8 bits	0x2: 4W1L	0x07: 8 bits
		1	A0 + 9b on 2 lanes	0x2: 32 bits	0x3: 4W2L	0x11: 18 bits
RGB888	24b	3	A0 + 8b on 1 lane	0x0: 8 bits	0x1: 3W1L	0x07: 8 bits
		1	A0 + 24b on 1 lane	0x2: 32 bits	0x1: 3W1L	0x17: 24 bits
		1	A0 + 12b on 2 lanes	0x2: 32 bits	0x3: 4W2L	0x17: 24 bits

8.5.1.4 Registers

8.5.1.4.1 CTRL1

- **Name:** DSPI Control Register 1

- **Description:** This register specifies the configuration parameters of DSPI.
- **Base Address:** 0x4002E000
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 8-222 DSPI Control Register 1

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15	BIDIMODE	RW	0x0	Bidirectional data mode enable. This bit enables half-duplex communication using common single bidirectional data line. Keep RXONLY bit clear when bidirectional mode is active. • 0x0: 2-line unidirectional data mode selected • 0x1: 1-line bidirectional data mode selected
14	BIDIOE	RW	0x0	Output enable in bidirectional mode. This bit combined with the BIDIMODE bit selects the direction of transfer in bidirectional mode. • 0x0: Output disabled (receive-only mode) • 0x1: Output enabled (transmit-only mode) **Note:** This bit applies to MOSI pin.
13	FPRST	RW	0x0	FIFO pointer reset This bit resets the pointers for TX FIFO and RX FIFO, works as a flush signal for both FIFOs. Any status flags related to RX FIFO and TX FIFO are also reset, as a result. • 0x0: No action. • 0x1: Flush TX FIFO and RX FIFO. **Note:** After this bit is written with 1 to flush the FIFOs, it should be written with 0 to ensure normal operation.
12:10	RSVD	R		Reserved bits
9	SSM	RW	0x0	Software slave management When the SSM bit is set, the CS pin input is replaced with the value from the SSI bit. • 0x0: Software slave management disabled

Bits	Field Name	RW	Reset	Description
				• 0x1: Software slave management enabled
8	SSI	RW	0x0	Internal slave select This bit takes effect only when the SSM bit is set. The value of this bit is forced onto the CS pin and the I/O. Value of the CS pin is ignored.
7	RSVD	R		Reserved bit
6	EN	RW	0x0	SPI enable • 0x0: Peripheral disabled • 0x1: Peripheral enabled
5:3	BR	RW	0x0	Baud rate control • 0x0: fPCLK/2 • 0x1: fPCLK/4 • 0x1: fPCLK/8 • 0x3: fPCLK/16 • 0x4: fPCLK/32 • 0x5: fPCLK/64 • 0x6: fPCLK/128 • 0x7: fPCLK/256 **Note:** These bits should not be changed when communication is ongoing.
2	MSTR	RW	0x0	Master selection • 0x0: Slave configuration • 0x1: Master configuration **Note:** This bit should not be changed when communication is ongoing.
1	CPOL	RW	0x0	Clock polarity • 0x0: CK to 0 in idle state. • 0x1: CK to 1 in idle state. **Note:** This bit should not be changed when communication is ongoing.

Bits	Field Name	RW	Reset	Description
				For DSPI operation adopting MIPI DBI Type-C, only CPOL=1, CPHA=1 combination is relevant; other combinations should not be used.
0	CPHA	RW	0x0	Clock phase • 0x0: The first clock transition is the first data capture edge • 0x1: The second clock transition is the first data capture edge **Note:** This bit should not be changed when communication is ongoing. For DSPI operation adopting MIPI DBI Type-C, only CPOL=1, CPHA=1 combination is relevant; other combinations should not be used.

8.5.1.4.2 CTRL2

- **Name:** DSPI Control Register 2
- **Description:** This register specifies the configuration parameters of DSPI.
- **Base Address:** 0x4002E000
- **Offset:** 0x04
- **Reset Value:** 0x0700

Table 8-223 DSPI Control Register 2

Bits	Field Name	RW	Reset	Description
31:13	RSVD	R		Reserved bits
12:8	DS	RW	0x07	Data size These 5 bits configure the data length for SPI transfers: • 0x0-0x2: Not used • 0x3-0x1F: 4 bits – 32 bits For regular SPI transfers, typical DS values are 8-bit, 16-bit, and 32-bit. For display SPI transfers, typical DS values are 8-bit, 9-bit, 16-bit, 18-bit, and 24-bit. **Note:**

Bits	Field Name	RW	Reset	Description
				If software attempts to write one of the “Not used” values, they are forced to be the value “0111” (8-bit).
7	TXEIE	RW	0x0	TX buffer empty interrupt enable • 0x0: TXE interrupt masked • 0x1: TXE interrupt not masked. Used to generate an interrupt request when the TXE flag is set.
6	RSVD	R		Reserved bit
5	ERRIE	RW	0x0	Error interrupt enable This bit controls the generation of an interrupt when an error condition occurs (OVR, MODF in SPI mode). • 0x0: Error interrupt is masked • 0x1: Error interrupt is enabled **Note:** This error will not occur in the current DSPI application.
4	FRF	RW	0x0	Frame format • 0x0: SPI Motorola mode • 0x1: SPI TI mode **Note:** This bit must be 0 at all times.
3	CSP	RW	0x0	CS pulse management This bit is used in master mode only. It allows the SPI to generate a CS pulse between two pieces of consecutive data in continuous transfers. In the case of a single data transfer, it forces the CS pin high level after the transfer. It has no meaning if CPHA = ’1’, or FRF = ’1’. • 0x0: No CS pulse • 0x1: CS pulse generated **Note:** This bit must be written only when the SPI is disabled (SPE=0).
2	SSOE	RW	0x0	SS output enable

Bits	Field Name	RW	Reset	Description
				• 0x0: SS output is disabled in master mode and the SPI interface can work in multimaster configuration. • 0x1: SS output is enabled in master mode and when the SPI interface is enabled. The SPI interface cannot work in a multimaster environment. **Note:** This bit must be 1 at all times.
1	TXDMAEN	RW	0x0	Tx buffer DMA enable When this bit is set, a DMA request is generated whenever the TXE flag is set. • 0x0: Tx buffer DMA disabled • 0x1: Tx buffer DMA enabled
0	RSVD	R		Reserved bit

8.5.1.4.3 STAT

- **Name:** DSPI Status Register
- **Description:** This register indicates the current working status of DSPI.
- **Base Address:** 0x4002E000
- **Offset:** 0x08
- **Reset Value:** 0x0002

Table 8-224 DSPI Status Register

Bits	Field Name	RW	Reset	Description
31:17	RSVD	R	0x0	Reserved bits
16:13	FTLCNT	RW	0x0	TX FIFO count in units of bytes These bits are set and cleared by hardware. • 0x0: FIFO empty • 0x1: 1/4 FIFO • 0x2: 1/2 FIFO • 0x3: FIFO full (considered as FULL when the FIFO threshold is greater than 1/2)
12:9	RSVD	R		Reserved bits
8	FRE	RW	0x0	Frame format error This flag is used for SPI in TI slave mode and I2S slave mode. [Not relevant]

Bits	Field Name	RW	Reset	Description
				This flag is set by hardware and reset when DSPI_SR is read by software. • 0x0: No frame format error • 0x1: A frame format error occurred
7	BUSY	RW	0x0	Busy flag • 0x0: SPI not busy • 0x1: SPI is busy in communication or Tx buffer is not empty This flag is set and cleared by hardware.
6	OVR	RW	0x0	Overrun flag • 0x0: No overrun occurred • 0x1: Overrun occurred This flag is set by hardware and reset by a software sequence. **Note:** This error will not occur in the current DSPI application.
5	MODF	RW	0x0	Mode fault • 0x0: No mode fault occurred • 0x1: Mode fault occurred This flag is set by hardware and reset by a software sequence. **Note:** This error will not occur in the current DSPI application.
4:2	RSVD	R		Reserved bits
1	TXE	RW	0x0	Transmit buffer empty • 0x0: Tx buffer not empty • 0x1: Tx buffer empty
0	RSVD	R		Reserved bit

8.5.1.4.4 DATA

- **Name:** DSPI Data Register
- **Description:** This register is a write data register.
- **Base Address:** 0x4002E000

- **Offset:** 0x0C
- **Reset Value:** 0x0000

Table 8-225 DSPI Data Register

Bits	Field Name	RW	Reset	Description
31:0	DATA	RW	0x0	Data register Data to be transmitted **Note:** Data is always right-aligned. Unused bits are ignored when writing to the register.

8.5.1.4.5 MODE

- **Name:** DSPI Display Mode Register
- **Description:** This register specifies the working mode parameters of DSPI.
- **Base Address:** 0x4002E000
- **Offset:** 0x10
- **Reset Value:** 0x0000

Table 8-226 DSPI Display Mode Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	DCX	RW	0x0	Value of data-versus-command information used in Display SPI Protocol Modes. This bit specifies the value to put on the DCX wire of Display SPI interface or the “A0” value to use on the DOUT bit stream when this information is provided as part of the outgoing data bits. • 0x0: Sets the data-versus-command value to 0. • 0x1: Sets the data-versus-command value to 1.
1:0	SPIMODE	RW	0x0	SPI protocol mode This bit selects the SPI protocol mode. • 0x0: Normal SPI (Motorola protocol only, not used) • 0x1: Display SPI 3-wire 1-lane interface • 0x2: Display SPI 4-wire 1-lane interface • 0x3: Display SPI 4-wire 2-lane interface

8.5.2 QSPI

8.5.2.1 Introduction

The GR5526 SoC supports three instances of Quad Serial Peripheral Interface (QSPI) master, namely QSPI M0, QSPI M1, and QSPI M2, which are used to interface to external slave devices. All the QSPI instances support memory mapped mode that can be used in accessing a storage device such as NOR Flash or PSRAM.

Each QSPI instance can be configured to work with standard SPI/Dual SPI/Quad SPI mode.

8.5.2.2 Main Features

- Standard/Dual/Quad Motorola SPI support
- Single slave select line to support 1 slave
- Built-in 16-/32-word RX/TX FIFOs for different instances
- Transfer data size up to 32 bits
- Programmable output interface frequency up to 48 MHz
- Supporting four SPI modes for different clock edge and phase configurations
- DMA support
- Access mode support:
  - Read and write transfer support in register mode
  - Read and write transfer support in memory mapped (XIP) mode
  - Supporting free data endian mode in XIP operation (for reads)
  - Continuous transfer mode in XIP operation (for reads)
  - Data prefetch support in XIP operation (for reads)
- Enhanced (Dual/Quad) SPI features:
  - Programmable address, instruction, wait cycles, and data frame size
  - Clock stretch feature support
- Programmable receive sample delay support
- Maskable interrupt generation
- Supporting adjusting the CS setup delay time
- Perfect access thoughtput

8.5.2.3 Functional Description

8.5.2.3.1 Block Diagram

- AHB interface and DMA controller interface
- Transmit and receive FIFO controllers and an FSM controller
- Register block
- Shift control and interrupt logic

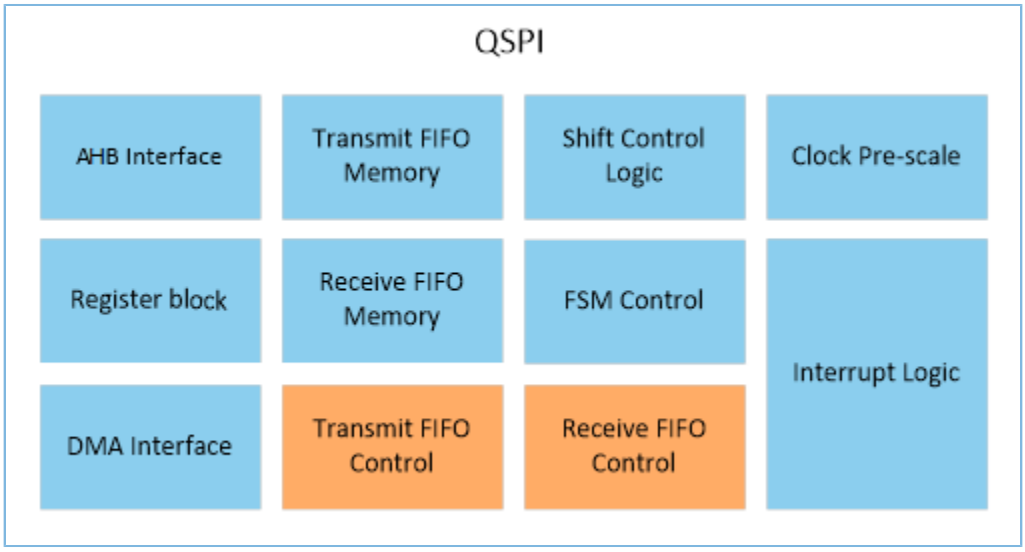

Figure 8-52 QSPI block diagram

8.5.2.3.2 Clock Ratios

The maximum frequency of the bit-rate clock (sclk_out) is one-half the frequency of ssi_clk signal. This is to allow the shift control logic to capture data on one clock edge of sclk_out signal and propagate data on the opposite edge; this is illustrated in the following figure.

The frequency of sclk_out signal can be derived from the following equation.

$$F_{sclk_{out}} = \frac{F_{ssi_{clk}}}{SCKDV} \tag{22}$$

Where, SCKDV (Baud Rate Register, BAUD[15:0]) is a programmable register holding any even value in the range of 0–65,534. If SCKDV = 0, sclk_out is disabled.

The sclk_out line only toggles when an active transfer is in progress. At all other time, it is held in inactive state, as define by the serial protocol under which it operates.

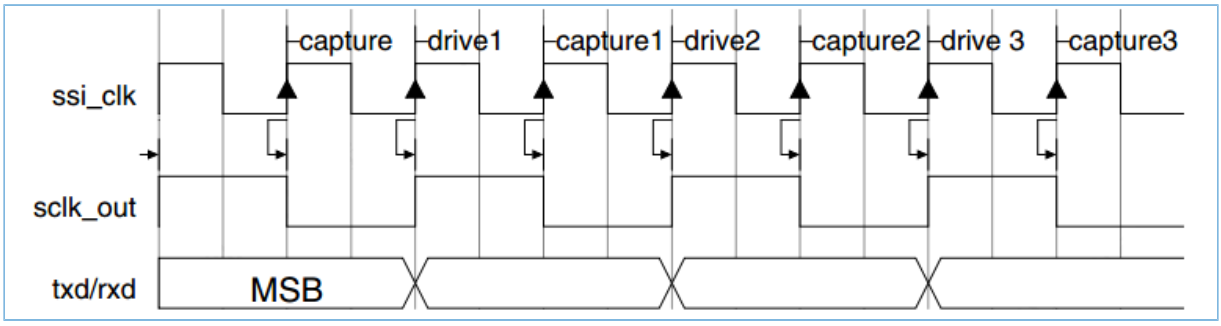

Figure 8-53 Maximum sclk_out/ssi_clk ratio

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

8.5.2.3.3 Motorola Serial Peripheral Interface

For details of the interface, see Section 8.4.2.4.1 Motorola Serial Peripheral Interface.

8.5.2.3.4 Transmit and Receive FIFO Buffers

The width of both TX and RX FIFO buffers is fixed at 32 bits. Data frames that are less than 32 bits must be right-justified when written into the TX FIFO buffer. The shift control logic automatically right-justifies receive data in the RX FIFO buffer.

Each data entry in the FIFO buffers contains a single data frame. Storing multiple data frames in a single FIFO location is not supported; for example, users may not store two 8-bit data frames in a single FIFO location. If an 8-bit data frame is required, the upper 8-bits of the FIFO entry are ignored or unused when the serial shifter transmits the data.

Transmit FIFO

The TX FIFO is loaded by AHB write commands to the QSPI data register (DATA). Data is popped (removed) from the TX FIFO by the shift control logic into the transmit shift register. The TX FIFO generates a FIFO empty interrupt request (qspi_txe_intrl) when the number of entries in the FIFO is less than or equal to the FIFO threshold value. The threshold value, set through the programmable transmit FIFO threshold level register (TX_FTL[4:0]), determines the level of FIFO entries at which an interrupt is generated. The threshold value allows users to provide early indication to the processor that the TX FIFO is nearly empty. A TX FIFO overflow interrupt (qspi_txo_intr) is generated if users attempt to write data into an already full TX FIFO.

Receive FIFO

Data is popped from the RX FIFO by AHB read commands to the QSPI data register (DATA). The RX FIFO is loaded from the receive shift register by the shift control logic. The RX FIFO generates a FIFO-full interrupt request (qspi_rxf_intr) when the number of entries in the FIFO is greater than or equal to the FIFO threshold value plus 1. The threshold value, set through programmable receive FIFO threshold level register (RX_FTL[4:0]), determines the level of FIFO entries at which an interrupt is generated.

The threshold value allows users to provide early indication to the processor that the RX FIFO is nearly full. A RX FIFO overrun interrupt (qspi_rxo_intr) is generated when the receive shift logic attempts to load data into a completely full RX FIFO. However, this newly received data is lost. A RX FIFO underflow interrupt (qspi_rxu_intr) is generated if users attempt to read from an empty RX FIFO. This alerts the processor that the read data is invalid.

8.5.2.3.5 QSPI Interrupts

The QSPI controller supports individual interrupt requests, each of which can be masked.

The QSPI controller interrupts are described as follows:

- TX FIFO empty interrupt (qspi_txe_intr) – Set when the TX FIFO is equal to or below its threshold value and requires service to prevent an under-run. The threshold value, set through a software-programmable register, determines the level of TX FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data is written into the TX FIFO buffer, bringing it over the threshold level.
- TX FIFO overflow interrupt (qspi_txo_intr) – Set when an AHB access attempts to write into the TX FIFO after it has been completely filled. When set, data written from the AHB is discarded. This interrupt remains set until users read the TX FIFO error interrupt clear register (TXEICR[0]).

- TX FIFO underflow interrupt (qspi_txu_intr) - Present only when internal DMA mode or SPI bridge mode or XIP write transfer is enabled. The interrupt is set when TX FIFO pop happens from serial FSM and FIFO is empty. This interrupt remains set until users read the TX FIFO error interrupt clear register (TXEICR[0]).
- RX FIFO full interrupt (qspi_rxf_intr) – Set when the RX FIFO is equal to or above its threshold value plus 1 and requires service to prevent an overflow. The threshold value, set through a software-programmable register, determines the level of RX FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data is read from the RX FIFO buffer, bringing it below the threshold level.
- RX FIFO overflow interrupt (qspi_rxo_intr) – Set when the receive logic attempts to place data into the RX FIFO after it has been completely filled. When set, newly received data is discarded. This interrupt remains set until users read the RX FIFO overflow interrupt clear register (RXOICR[0]).
- RX FIFO underflow interrupt (qspi_rxu_intr) – Set when an AHB access attempts to read from the RX FIFO when it is empty. When set, zeros are read back from the RX FIFO. This interrupt remains set until users read the RX FIFO underflow interrupt clear register (RXUICR[0]).

8.5.2.3.6 Transfer Modes

When transferring data on the serial bus, the SPI controller operates in the modes discussed in this section. The transfer mode (TMOD) is set by writing to bit8 and bit9 of the SPI control register 0 (CTRL0).

Note:

The transfer mode setting does not affect the duplex of the serial transfer.

8.5.2.3.6.1 Transmit and Receive

When CTRL0[11:10] = 0x0, both transmit and receive logics are valid. The data transfer occurs as normal according to the selected frame format (serial protocol). Transmit data is popped from the TX FIFO and sent through the TXD line to the target device, which replies with data on the RXD line. The receive data from the target device is moved from the receive shift register into the RX FIFO at the end of each data frame.

8.5.2.3.6.2 Transmit Only

When CTRL0[11:10] = 0x1, the received data is invalid and should not be stored in the RX FIFO. The data transfer occurs as normal, according to the selected frame format (serial protocol). Transmit data is popped from the TX FIFO and sent through the TXD line to the target device, which replies with data on the RXD line. At the end of the data frame, the receive shift register does not load its newly received data into the RX FIFO. The data in the receive shift register is overwritten by the next transfer. Users should mask interrupts originating from the receive logic when this mode is entered.

8.5.2.3.6.3 Receive Only

When CTRL0[11:10] = 0x2, the transmit data is invalid. When QSPI is configured as a slave, the TX FIFO is never popped in Receive Only mode. The TXD output remains at a constant logic level during the transmission. The data transfer occurs as normal according to the selected frame format (serial protocol). The receive data from the target device is

moved from the receive shift register into the RX FIFO at the end of each data frame. Users should mask interrupts originating from the transmit logic when this mode is entered.

8.5.2.3.6.4 EEPROM Read

When CTRL0[11:10] = 0x3, the transmit data is used to transmit an opcode and/or an address to the EEPROM device. Typically, this takes three data frames (8-bit opcode followed by 8-bit upper address and 8-bit lower address). During the transmission of the opcode and address, no data is captured by the receive logic (as long as the SPI controller master is transmitting data on its TXD line, data on the RXD line is ignored). The QSPI controller master continues to transmit data until the TX FIFO is empty. Therefore, users should only have enough data frames in the TX FIFO to supply the opcode and address to the EEPROM. If more data frames are in the TX FIFO than needed, read data is lost.

When the TX FIFO becomes empty (all control information has been sent), data on the receive line (RXD) is valid and is stored in the RX FIFO; the TXD output is held at a constant logic level. The serial transfer continues until the number of data frames received by the QSPI controller master matches the value of the NDF (Number of Data Frames) field in the SPI Control register 1 (CTRL1[15:0]) + 1.

8.5.2.3.7 DMA

In the register mode (non-XIP mode), QSPI uses a handshake request signal to start the DMA transfer. DMA0 instance provides handshake request signals to QSPI0 and QSPI1 instances, and DMA1 instance provides handshake request signals to QSPI1 and QSPI2 instances, so

- QSPI M0 can use DMA0 to receive and transmit data.
- QSPI M1 can use DMA0/DMA1 to receive and transmit data.
- QSPI M2 can use DMA1 to receive and transmit data.

Meanwhile, Channel0 and Channel1 of DMA have the biggest FIFO depth. It is suggested to assign the Channel0 & Channel1 of DMA to QSPI instances to get the maximum throughput.

8.5.2.3.8 Clock Stretching in Enhanced SPI Transfers

While QSPI is transmitting or receiving data from a QSPI slave device, the software may not be able to keep up with the transfer rate due to the bandwidth issues on the slave interface. In such cases, TX FIFO becomes empty or RX FIFO overflows while transmitting and receiving the data.

To avoid data corruption, CPU must discard the current operation and start a new transfer. This process is time consuming and inefficient.

To handle such scenarios, clock stretching support has been added in QSPI.

- For write transactions, whenever TX FIFO becomes empty, QSPI does not de-select the slave. Instead, it masks the clock until the new data is pushed into the TX FIFO. Hence, the transfer is not broken, and CPU intervention is not required. The transfer is resumed after the TX FIFO receives at least one FIFO entry.

- For read transactions, similar to the write transaction, whenever QSPI detects that RX FIFO is full, the clock is masked until the data is read from FIFO. The transfer is resumed when FIFO level goes below the set threshold level in the RX_FTL[4:0] register.

  QSPI block provides a programming bit (CLK_STRETCH_EN) in QSPI enhanced control register 0 (SPI_CTRL0[30]) to enable clock stretching feature.

8.5.2.3.9 Memory Mapped Mode (XIP Mode)

QSPI module allows directly performing memory read operations from AHB transaction. This is called memory mapped access mode (also called XIP mode), in which QSPI acts as memory mapped interface to an SPI memory.

8.5.2.3.9.1 Memory Mapped Read Model

XIP read operation works under memory mapped mode. Typical devices are NOR Flash and PSRAM with QSPI/DSPI interface.

XIP operations are supported only in Dual or Quad enhanced SPI modes of operation, so bit (SPI_FRF) of the CTRL0[23:22] must not be programmed to 0. When CTRL0[23:22] = 0x1, select Dual SPI frame format. When CTRL0[23:22] = 0x2, select Quad SPI frame format. Typically, an XIP operation consists of an address phase and a data phase. The programming flow to set up an XIP transfer is as follows:

- Set the SPI protocol format value to Motorola format in CTRL0.FRF register.
- Set the SPI frame format value in XIP_CTRL.FRF register.
- Set the address length, wait cycles, and transaction type in XIP_CTRL.ADDR_L register. It should be noted that the maximum address length is 32 bits.

The instruction phase can also be included in XIP transfer by using INST_EN field of the XIP_CTRL register. In this case, the following registers must be set:

- Set length of instruction in the INST_L field of XIP_CTRL[10:9] register.
- Write the instruction opcodes in the XIP_INCR_INST[15:0] and XIP_WRAP_INST[15:0] registers.

After the programming is complete, users can initiate a read transaction through the AHB interface that is transferred to the SPI peripheral using programmed values.

After finishing above setting, users can read the device's data by accessing the corresponding QSPI's memory mapped space.

8.5.2.3.9.2 Memory Mapped Write Model

XIP write feature can be used to access PSRAM device supporting QSPI/DSPI interface (it cannot be used to write the NOR Flash device). Each typical write time meets with the PSRAM's tCEM time.

XIP operations are supported only in enhanced SPI modes of operation (Dual and Quad modes).

QSPI can write into the DSPI/QSPI interface memory without software overhead of writing the instruction and address separately into the transmit buffer. XIP write feature executes the write transactions directly without software intervention.

The programming flow to set up an XIP transfer is as follows:

- Set the SPI frame format value in XIP_WR_CTRL. WR_CTRL_FRF register.
- Set the address length, wait cycles, and transaction type in XIP_WR_CTRL.WR_CTRL_ADDRL. It should be noted that the maximum address length is 32 bits.
- Set length of instruction in the XIP_WR_CTRL.WR_INST_L register.
- Write the instruction opcodes in the XIP_WR_INCR_INST and XIP_WR_WRAP_INST registers.

8.5.2.3.9.3 Memory Mapped Space

After activating the memory mapped mode, users can perform access operations from AHB transaction. The AHB memory space for QSPI instances are allocated as Figure 8-54.

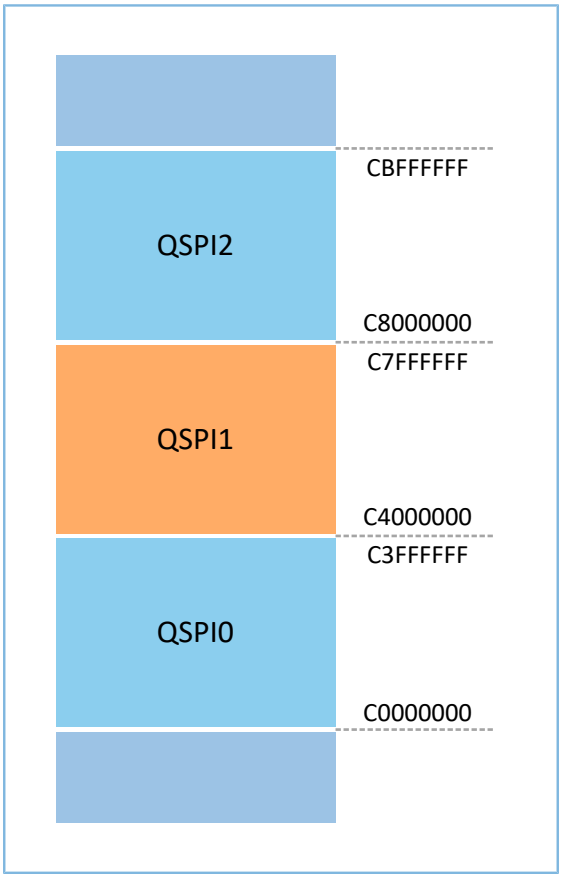

Figure 8-54 Memory map address for QSPI

- QSPI0 memory mapped space is from C0000000 to C3FFFFFF; 65,536 KB in total.
- QSPI1 memory mapped space is from C4000000 to C7FFFFFF; 65,536 KB in total.
- QSPI2 memory mapped space is from C8000000 to CBFFFFFF; 65,536 KB in total.

8.5.2.3.9.4 Data Endian Mode for Memory Mapped Read Model

QSPI supports different endian data in XIP Read mode. Users can refer to the following chart to configure the endian mode as needed. The related register is QSPI_MODE_XIP register.

Table 8-227 Different endian data in XIP read mode

AHB Address	0xAC000000	0xAC000001	0xAC000002	0xAC000003	0xAC000004	0xAC000005	0xAC000006	0xAC000007	0xAC000008	0xAC000009	0xAC00000A	0xAC00000B
PSRAM Address	0x000000	0x000001	0x000002	0x000003	0x000004	0x000005	0x000006	0x000007	0x000008	0x000009	0x00000A	0x00000B
Saved Data	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C

Table 8-228 XIP read by different mode & address type

Data Type	By Byte				By Short		By Integer	
Access Addr/ Endian Mode	0xAC000000	0xAC000001	0xAC000002	0xAC000003	0xAC000000	0xAC000002	0xAC000000	0xAC000004
Mode 0/3	0x01	0x02	0x03	0x04	0x0102	0x0304	0x01020304	0x05060708
Mode 1	0x01	0x02	0x03	0x04	0x0201	0x0403	0x02010403	0x06050807
Mode 2	0x01	0x02	0x03	0x04	0x0201	0x0403	0x04030201	0x08070605

8.5.2.3.10 Enhanced Feature

8.5.2.3.10.1 Related Delay Between CS and SCK

T_{csu} (CS Setup Delay Time) is an important parameter to access QSPI device. If T_{csu} is not enough, the access may fail. T_{ch} (CS Release Delay Time) is symmetric to T_{csu}.

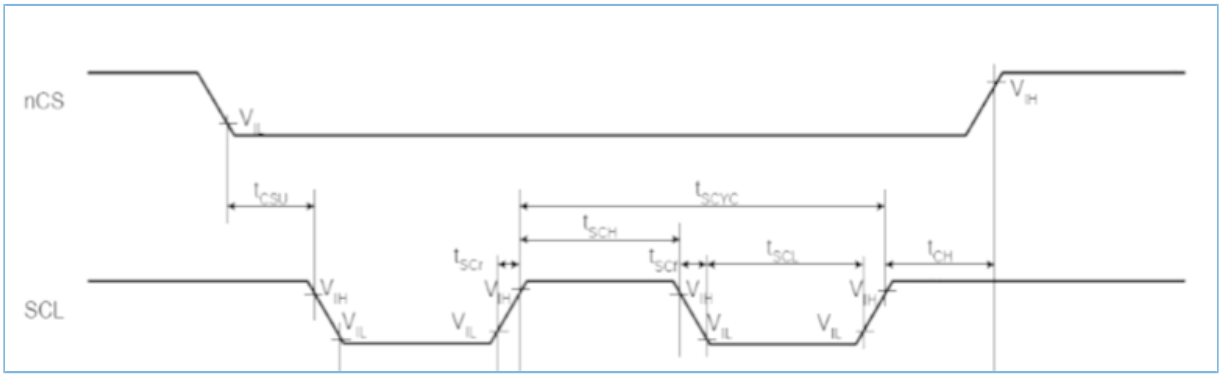

Figure 8-55 Relationship between T_{ch} and T_{csu}

If necessary, users can set the two parameters wider to fit with devices' requirement. The related registers are QSPI_CS_SETUP_DLY and QSPI_CS_RELEASE_DLY.

8.5.2.4 Registers

8.5.2.4.1 CTRL0

- **Name:** QSPI Control Register 0
- **Description:** This register controls the serial data transfer. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x00
- **Reset Value:** 0x01070000

Table 8-229 QSPI Control Register 0

Bits	Field Name	RW	Reset	Description
31:24	RSVD	R		Reserved bits
23:22	SPI_FRF	RW	0x0	QSPI frame format: Selects data frame format for transmitting/receiving the data bits. **Value:** • 0x0 (STD_SPI_FRF): Standard SPI frame format • 0x1 (DUAL_SPI_FRF): Dual SPI frame format • 0x2 (QUAD_SPI_FRF): Quad SPI frame format • 0x3: Reserved
21:15	RSVD	R		Reserved bits
14	SSTEN	RW	0x0	Slave select toggle enable. When operating in SPI mode with clock phase (SCPH) set to 0, this register controls the behavior of the slave select line (CS) between data frames. If this register field is set to 1, the CS line will toggle between consecutive data frames, with the serial clock (SCLK) being held to its default value while CS is high; if this register field is set to 0, the CS will stay low and SCLK will run continuously for the duration of the transfer. **Note:** When the SPI is configured as a slave, this register serves no purpose.
13	SRL	RW	0x0	Shift register loop. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (NORMAL_MODE): Normal mode operation • 0x1 (TESTING_MODE): Test mode: TX & RX shift reg connected
12	SLVOE	RW	0x0	Slave output enable. **Value:** • 0x0 (ENABLED): Slave output is enabled. • 0x1 (DISABLED): Slave output is disabled.
11:10	TMOD	RW	0x0	Transfer mode. This transfer mode is only valid when the SPI is configured as master device. **Value:** • 0x0 (TX_AND_RX): Transmit & receive • 0x1 (TX_ONLY): Transmit only mode • 0x2 (RX_ONLY): Receive only mode • 0x3 (EEPROM_READ): EEPROM read mode
9	SCPOL	RW	0x0	Serial clock polarity. Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the SPI master is not actively transferring data on the serial bus. **Value:** • 0x0 (SCLK_LOW): Inactive state of serial clock is low • 0x1 (SCLK_HIGH): Inactive state of serial clock is high
8	SCPHA	RW	0x0	Serial clock phase. Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data is captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data is captured on the second edge of the serial clock. **Value:** • 0x0 (SCPH_MIDDLE): Serial clock toggles in middle of first data bit

Bits	Field Name	RW	Reset	Description
				• 0x1 (SCPH_START): Serial clock toggles at start of first data bit
7:6	FRF	RW	0x0	Frame format. Selects which serial protocol transfers the data. **Value:** • 0x0 (MOTOROLA_SPI): Motorolla SPI frame format • 0x1 (TEXAS_SSP): Texas Instruments SSP frame format • 0x2 (NS_MICROWIRE): National Microwire frame format • 0x3 (RESERVED): Reserved value
5	RSVD	R		Reserved bits
4:0	DFS	RW	0x7	Data frame size in 32-bit transfer size mode. Used to select the data frame size in 32-bit transfer mode. When the data frame size is programmed to be less than 32 bits, the receive data is automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. Users are responsible for making sure that transmit data is right-justified before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. **Value:** • 0x3 (FRAME_04BITS): 4-bit serial data transfer • 0x4 (FRAME_05BITS): 5-bit serial data transfer • 0x5 (FRAME_06BITS): 6-bit serial data transfer • 0x6 (FRAME_07BITS): 7-bit serial data transfer • 0x7 (FRAME_08BITS): 8-bit serial data transfer • 0x8 (FRAME_09BITS): 9-bit serial data transfer • 0x9 (FRAME_10BITS): 10-bit serial data transfer • 0xa (FRAME_11BITS): 11-bit serial data transfer • 0xb (FRAME_12BITS): 12-bit serial data transfer • 0xc (FRAME_13BITS): 13-bit serial data transfer • 0xd (FRAME_14BITS): 14-bit serial data transfer • 0xe (FRAME_15BITS): 15-bit serial data transfer • 0xf (FRAME_16BITS): 16-bit serial data transfer • 0x10 (FRAME_17BITS): 17-bit serial data transfer

Bits	Field Name	RW	Reset	Description
				• 0x11 (FRAME_18BITS): 18-bit serial data transfer • 0x12 (FRAME_19BITS): 19-bit serial data transfer • 0x13 (FRAME_20BITS): 20-bit serial data transfer • 0x14 (FRAME_21BITS): 21-bit serial data transfer • 0x15 (FRAME_22BITS): 22-bit serial data transfer • 0x16 (FRAME_23BITS): 23-bit serial data transfer • 0x17 (FRAME_24BITS): 24-bit serial data transfer • 0x18 (FRAME_25BITS): 25-bit serial data transfer • 0x19 (FRAME_26BITS): 26-bit serial data transfer • 0x1a (FRAME_27BITS): 27-bit serial data transfer • 0x1b (FRAME_28BITS): 28-bit serial data transfer • 0x1c (FRAME_29BITS): 29-bit serial data transfer • 0x1d (FRAME_30BITS): 30-bit serial data transfer • 0x1e (FRAME_31BITS): 31-bit serial data transfer • 0x1f (FRAME_32BITS): 32-bit serial data transfer

8.5.2.4.2 CTRL1

- **Name:** QSPI Control Register 1
- **Description:** Control register 1 controls the end of serial transfers in receive only mode. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x04
- **Reset Value:** 0x00000000

Table 8-230 QSPI Control Register 1

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	NDF	RW	0x0	Number of data frames. When CTRL0[11:10] = 0x2 or CTRL0[11:10] = 0x3, this register field sets the number of data frames to be continuously received by the QSPI. The QSPI continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables users to receive up to 64 KB of data in a continuous transfer.

8.5.2.4.3 QSPI_EN

- **Name:** QSPI Enable Register
- **Description:** This register enables and disables the QSPI.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x08
- **Reset Value:** 0x00000000

Table 8-231 QSPI Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	SSI_EN	RW	0x0	QSPI enable. Enables and disables all QSPI operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. **Value:** • 0x0 (DISABLE): Disable serial transfer • 0x1 (ENABLED): Enable serial transfer

8.5.2.4.4 MWC

- **Name:** Microwire Control Register
- **Description:** This register controls the direction of the data word for the half-duplex Microwire serial protocol. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x0C
- **Reset Value:** 0x00000000

Table 8-232 Microwire Control Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	MHS	RW	0x0	Microwire handshaking. Used to enable and disable the busy/ready handshaking interface for the Microwire protocol. When this bit is enabled, the QSPI checks for a ready status from the target slave, after the transfer of the last data/control bit, before clearing the BUSY status in the STAT register.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (DISABLE): Handshaking interface is disabled • 0x1 (ENABLED): Handshaking interface is enabled
1	MDD	RW	0x0	Microwire control. Defines the direction of the data word when the Microwire serial protocol is used. **Value:** • 0x0 (RECEIVE): QSPI receives data • 0x1 (TRANSMIT): QSPI transmits data
0	MWMOD	RW	0x0	Microwire transfer mode. Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. **Value:** • 0x0 (NON_SEQUENTIAL): Non-sequential Microwire transfer • 0x1 (SEQUENTIAL): Sequential Microwire transfer

8.5.2.4.5 SE

- **Name:** Slave Enable Register
- **Description:** The register enables the individual slave select output lines from the QSPI master.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 8-233 Slave Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	SLAVE0	RW	0x0	Slave select enable. Each bit in this register corresponds to a slave select line (CSn) from the QSPI master. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (NOT_SELECTED): No slave selected • 0x1 (SELECTED): Slave is selected **Note:** When the QSPI is configured as a slave, this register serves no purpose.

8.5.2.4.6 BAUD

- **Name:** Baud Rate Register
- **Description:** The register derives the frequency of the serial clock that regulates the data transfer. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 8-234 Baud Rate Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:0	SCLKDIV	RW	0x0	QSPI clock divider. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: Fsclk_out = Fssi_clk/SCKDV

8.5.2.4.7 TX_FTL

- **Name:** Transmit FIFO Threshold Level Register
- **Description:** This register controls the threshold value for the transmit FIFO memory.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 8-235 Transmit FIFO Threshold Level Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
20:16	TXFTHR	RW	0x0	Transfer start FIFO level. Used to control the level of entries in transmit FIFO above which transfer will start on serial line. This register can be used to ensure that sufficient data is present in transmit FIFO before starting a write operation on serial line.
15:5	RSVD	R		Reserved bits
4:0	TFT	RW	0x0	Transmit FIFO threshold. Controls the level of entries at (or below) which the transmit FIFO controller triggers an interrupt. The FIFO depth is configurable in 16/32 words; this register is sized to the number of address bits needed to access the FIFO. If users attempt to set this value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

8.5.2.4.8 RX_FTL

- **Name:** Receive FIFO Threshold Level Register
- **Description:** This register controls the threshold value for the receive FIFO memory.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x1C
- **Reset Value:** 0x00000000

Table 8-236 Receive FIFO Threshold Level Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4:0	RFT	RW	0x0	Receive FIFO threshold. Controls the level of entries at (or above) which the receive FIFO controller triggers an interrupt. The FIFO depth is configurable in 16/32 words. This register is sized to the number of address bits needed to access the FIFO. If users

Bits	Field Name	RW	Reset	Description
				attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.

8.5.2.4.9 TX_FL

- **Name:** Transmit FIFO Level Register
- **Description:** This register contains the number of valid data entries in the transmit FIFO memory.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 8-237 Transmit FIFO Level Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5:0	TX_FL	R	0x0	Transmit FIFO level. Contains the number of valid data entries in the transmit FIFO.

8.5.2.4.10 RX_FL

- **Name:** Receive FIFO Level Register
- **Description:** This register contains the number of valid data entries in the receive FIFO memory. This register can be ready at any time.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 8-238 Receive FIFO Level Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5:0	RX_FL	R	0x0	Receive FIFO level. Contains the number of valid data entries in the receive FIFO.

8.5.2.4.11 STAT

- **Name:** Status Register
- **Description:** This is a read-only register used to indicate the current transfer status, FIFO status, and any transmission/reception errors that may have occurred. The status register may be read at any time.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x28
- **Reset Value:** 0x00000006

Table 8-239 Status Register

Bits	Field Name	RW	Reset	Description
31:7	RSVD	R		Reserved bits
6	DCOL	R	0x0	Data collision error. This bit will be set if MISO input is asserted by other master, when the QSPI master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read.
5	TXE	R	0x0	Transmission Error. Set if the transmit FIFO is empty when a transfer is started. Data from the previous transmission is resent on the TXD line. This bit is cleared when read. **Value:** • 0x0 (NO_ERROR): No error • 0x1 (TX_ERROR): Transmission error
4	RFF	R	0x0	Receive FIFO full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty locations, this bit is cleared. **Value:** • 0x0 (NOT_FULL): Receive FIFO is not full • 0x1 (FULL): Receive FIFO is full
3	RFNE	R	0x0	Receive FIFO not empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (EMPTY): Receive FIFO is empty • 0x1 (NOT_EMPTY): Receive FIFO is not empty
2	TFE	R	0x1	Transmit FIFO empty. When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. **Value:** • 0x0 (NOT_EMPTY): Transmit FIFO is not empty • 0x1 (EMPTY): Transmit FIFO is empty
1	TFNF	R	0x1	Transmit FIFO not full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. **Value:** • 0x0 (FULL): Transmit FIFO is full • 0x1 (NOT_FULL): Transmit FIFO is not Full
0	BUSY	R	0x0	QSPI busy flag. This bit being set indicates that a serial transfer is in progress; this bit being cleared indicates that the QSPI is idle or disabled. **Value:** • 0x0 (INACTIVE): QSPI is idle or disabled • 0x1 (ACTIVE): QSPI is actively transferring data

8.5.2.4.12 INTMASK

- **Name:** Interrupt Mask Register
- **Description:** This read/write reigster masks or enables all interrupts generated by the QSPI. When the QSPI is configured as a slave device, the MSTIM bit field is not present. This changes the reset value from 0x3F for serial-master configurations to 0x1F for serial-slave configurations.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x2C
- **Reset Value:** master ? 0x0000003F: 0x0000001F

Table 8-240 Interrupt Mask Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	MSTIM	RW	0x1	Multi-master contention interrupt mask. **Value:** • 0x0 (MASKED): Disable mst_intr interrupt • 0x1 (UNMASKED): Enable mst_intr interrupt
4	RXFIM	RW	0x1	Receive FIFO full interrupt mask **Value:** • 0x0 (MASKED): Disable rxf_intr interrupt • 0x1 (UNMASKED): Enable rxf_intr interrupt
3	RXOIM	RW	0x1	Receive FIFO overflow interrupt mask **Value:** • 0x0 (MASKED): Disable rxo_intr interrupt • 0x1 (UNMASKED): Enable rxo_intr interrupt
2	RXUIM	RW	0x1	Receive FIFO underflow interrupt mask **Value:** • 0x0 (MASKED): Disable rxu_intr interrupt • 0x1 (UNMASKED): Enable rxu_intr interrupt
1	TXOIM	RW	0x1	Transmit FIFO overflow interrupt mask **Value:** • 0x0 (MASKED): Disable txo_intr interrupt • 0x1 (UNMASKED): Enable txo_intr interrupt
0	TXEIM	RW	0x1	Transmit FIFO empty interrupt mask **Value:** • 0x0 (MASKED): Disable txe_intr interrupt • 0x1 (UNMASKED): Enable txe_intr interrupt

8.5.2.4.13 INTSTAT

- **Name:** Interrupt Status Register
- **Description:** This register reports the status of the QSPI interrupts after they have been enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x30
- **Reset Value:** 0x00000000

Table 8-241 Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	MSTIS	R	0x0	Multi-master contention interrupt status **Value:** • 0x0 (INACTIVE): mst_intr interrupt is not active after being enabled • 0x1 (ACTIVE): mst_intr interrupt is full after being enabled
4	RXFIS	R	0x0	Receive FIFO full interrupt status **Value:** • 0x0 (INACTIVE): rxf_intr interrupt is not active after being enabled • 0x1 (ACTIVE): rxf_intr interrupt is full after being enabled
3	RXOIS	R	0x0	Receive FIFO overflow interrupt status **Value:** • 0x0 (INACTIVE): rxo_intr interrupt is not active after being enabled • 0x1 (ACTIVE): rxo_intr interrupt is active after being enabled
2	RXUIS	R	0x0	Receive FIFO underflow interrupt status **Value:** • 0x0 (INACTIVE): rxu_intr interrupt is not active after being enabled • 0x1 (ACTIVE): rxu_intr interrupt is active after being enabled
1	TXOIS	R	0x0	Transmit FIFO overflow interrupt status **Value:** • 0x0 (INACTIVE): txo_intr interrupt is not active after being enabled • 0x1 (ACTIVE): txo_intr interrupt is active after being enabled
0	TXEIS	R	0x0	Transmit FIFO empty interrupt status **Value:** • 0x0 (INACTIVE): txe_intr interrupt is not active after being enabled • 0x1 (ACTIVE): txe_intr interrupt is active after being enabled

8.5.2.4.14 RAW_INTSTAT

- **Name:** Raw Interrupt Status Register
- **Description:** This read-only register reports the status of the QSPI interrupts prior to being enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x34
- **Reset Value:** 0x00000000

Table 8-242 Raw Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	MSTIR	R	0x0	Multi-master contention raw interrupt status **Value:** • 0x0 (INACTIVE): mst_intr interrupt is not active prior to being enabled • 0x1 (ACTIVE): mst_intr interrupt is full prior to being enabled
4	RXFIR	R	0x0	Receive FIFO full raw interrupt status **Value:** • 0x0 (INACTIVE): rxf_intr interrupt is not active prior to being enabled • 0x1 (ACTIVE): rxf_intr interrupt is active prior to being enabled
3	RXOIR	R	0x0	Receive FIFO overflow raw interrupt status **Value:** • 0x1 (ACTIVE): rxo_intr interrupt is not active prior to being enabled • 0x0 (INACTIVE): rxo_intr interrupt is active prior to being enabled
2	RXUIR	R	0x0	Receive FIFO underflow raw interrupt status **Value:** • 0x0 (INACTIVE): rxu_intr interrupt is not active prior to being enabled • 0x1 (ACTIVE): rxu_intr interrupt is active prior to being enabled
1	TXOIR	R	0x0	Transmit FIFO overflow raw interrupt status **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0 (INACTIVE): txo_intr interrupt is not active prior to being enabled • 0x1 (ACTIVE): txo_intr interrupt is active prior to being enabled
0	TXEIR	R	0x0	Transmit FIFO empty raw interrupt status **Value:** • 0x0 (INACTIVE): txe_intr interrupt is not active prior to being enabled • 0x1 (ACTIVE): txe_intr interrupt is active prior to being enabled

8.5.2.4.15 TXEIC

- **Name:** Transmit FIFO Error Interrupt Clear Register
- **Description:** This register is used to clear the transmit FIFO error interrupt.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 8-243 Transmit FIFO Error Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	TXEIC	R	0x0	Clear transmit FIFO overflow/underflow interrupt. This register reflects the status of the interrupt. A read from this register clears the txo_intr interrupt; writing has no effect.

8.5.2.4.16 RXOIC

- **Name:** Receive FIFO Overflow Interrupt Clear Register
- **Description:** This register is used to clear the receive FIFO overflow interrupt.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x3C
- **Reset Value:** 0x00000000

Table 8-244 Receive FIFO Overflow Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RXOIC	R	0x0	Clear receive FIFO overflow interrupt. This register reflects the status of the interrupt. A read from this register clears the rxo_intr interrupt; writing has no effect.

8.5.2.4.17 RXUIC

- **Name:** Receive FIFO Underflow Interrupt Clear Register
- **Description:** This register is used to clear the receive FIFO underflow interrupt.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 8-245 Receive FIFO Underflow Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RXUIC	R		Clear receive FIFO underflow interrupt. This register reflects the status of the interrupt. A read from this register clears the rxu_intr interrupt; writing has no effect.

8.5.2.4.18 MSTIC

- **Name:** Multi-Master Interrupt Clear Register
- **Description:** This register is used to clear the multi-master interrupt.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x44
- **Reset Value:** 0x00000000

Table 8-246 Multi-Master Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	MSTIC	R		Clear multi-master contention interrupt.

Bits	Field Name	RW	Reset	Description
				This register reflects the status of the interrupt. A read from this register clears the mst_intr interrupt; writing has no effect.

8.5.2.4.19 INTCLR

- **Name:** Interrupt Clear Register
- **Description:** This register is used to clear interrupts.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x48
- **Reset Value:** 0x00000000

Table 8-247 Interrupt Clear Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	INTCLR	R		Clear interrupts. This register is set if any of the interrupts below are active. A read clears the txo_intr, rxu_intr, rxo_intr, and the mst_intr interrupts. Writing to this register has no effect.

8.5.2.4.20 DMAC

- **Name:** DMA Control Register
- **Description:** The register is used to enable the DMA Controller interface operation.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x4C
- **Reset Value:** 0x00000000

Table 8-248 DMA Control Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	TDMAE	RW		Transmit DMA enable. This bit enables/disables the TX FIFO DMA channel. **Value:** • 0x0 (DISABLE): Transmit DMA disabled • 0x1 (ENABLED): Transmit DMA enabled

Bits	Field Name	RW	Reset	Description
0	RDMAE	RW		Receive DMA enable. This bit enables/disables the receive FIFO DMA channel. **Value:** • 0x0 (DISABLE): Receive DMA disabled • 0x1 (ENABLED): Receive DMA enabled

8.5.2.4.21 DMA_TDL

- **Name:** DMA Transmit Data Level Register
- **Description:** This register controls the threshold value for the TX FIFO memory.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 8-249 DMA Transmit Data Level Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4:0	DMA_TDL	RW		Transmit data level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the TX FIFO is equal to or below this field value, and TDMAE = 1.

8.5.2.4.22 DMA_RDL

- **Name:** DMA Receive Data Level Register
- **Description:** This register controls the threshold value for the receive FIFO memory.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x54
- **Reset Value:** 0x00000000

Table 8-250 DMA Receive Data Level Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4:0	DMA_RDL	RW		Receive data level.

Bits	Field Name	RW	Reset	Description
				This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMA_RDL +1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMAE=1.

8.5.2.4.23 DATA

- **Name:** Data Register
- **Description:** The QSPI data register is a 32-bit read/write buffer for the TX/RX FIFOs. When the register is read, data in the RX FIFO buffer is accessed. When it is written to, data is moved into the TX FIFO buffer; a write can occur only when SSI_EN[0] = 1. FIFOs are reset when SSI_EN[0] = 0.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x60
- **Reset Value:** 0x00000000

Table 8-251 Data Register

Bits	Field Name	RW	Reset	Description
31:0	DATA	RW		Data register. When writing to this register, users must right-justify the data. Read data is automatically right-justified.

8.5.2.4.24 RX_SMP_DLY

- **Name:** Receive Sample Delay Register
- **Description:** This register controls the number of SCLK cycles that are delayed (from the default sample time) before the actual sample of the RXD input occurs. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0xF0
- **Reset Value:** 0x00000000

Table 8-252 Receive Sample Delay Register

Bits	Field Name	RW	Reset	Description
3:0	RX_SAMPLEDLY	RW		Receive sample delay

8.5.2.4.25 SPI_CTRL0

- **Name:** QSPI Enhanced Control Register 0
- **Description:** This register is used to control the serial data transfer in enhanced SPI mode of operation. The register is relevant only when CTRL0.SPI_FRF is set to either 01 or 10 or 11. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0xF4
- **Reset Value:** 0x00000000

Table 8-253 QSPI Enhanced Control Register 0

Bits	Field Name	RW	Reset	Description
31	RSVD	R		Reserved bits
30	CLK_STRETCH_EN	RW	0x0	Enables clock stretching capability in QSPI transfers. • In case of write operations, if the transmit FIFO becomes empty, DWC_ssi will stretch the clock until the FIFO has enough data to continue the transfer. • In case of read operations, if the receive FIFO becomes full, DWC_ssi will stop the clock until data has been read from the FIFO.
29:16	RSVD	R		Reserved bits
15:11	WAITCYCLES	RW	0x0	Wait cycles in Dual/Quad mode between control frames transmit and data reception. Specified as number of QSPI clock cycles.
10	RSVD	R		Reserved bit
9:8	INSTL	RW	0x0	Dual/Quad mode instruction length in bits. **Value:** • 0x0 (INST_L0): No instruction • 0x1 (INST_L4): 4-bit instruction length • 0x2 (INST_L8): 8-bit instruction length • 0x3 (INST_L16): 16-bit instruction length
7:6	RSVD	R		Reserved bits
5:2	ADDRL	RW	0x0	These bits define the length of address to be transmitted. Only after these bits are programmed into the FIFO, the transfer can begin. **Value:** • 0x0 (ADDR_L0): No address • 0x1 (ADDR_L4): 4-bit address length

Bits	Field Name	RW	Reset	Description
				• 0x2 (ADDR_L8): 8-bit address length • 0x3 (ADDR_L12): 12-bit address length • 0x4 (ADDR_L16): 16-bit address length • 0x5 (ADDR_L20): 20-bit address length • 0x6 (ADDR_L24): 24-bit address length • 0x7 (ADDR_L28): 28-bit address length • 0x8 (ADDR_L32): 32-bit address length
1:0	TRANSTYPE	RW	0x0	Address and instruction transfer format. Selects whether QSPI will transmit instruction/address either in standard SPI mode or the SPI mode selected in CTRL0.SPI_FRF field. **Value:** • 0x0: Instruction and address will be sent in standard SPI Mode. • 0x1: Instruction will be sent in standard SPI mode and address will be sent in the mode specified by CTRL0.SPI_FRF. • 0x2: Both instruction and address will be sent in the mode specified by SPI_FRF. • 0x3: Reserved.

8.5.2.4.26 XIP_MODE_BITS

- **Name:** XIP Mode Bits Register
- **Description:** This register carries the mode bits which are sent in the XIP mode of operation after address phase. This is an 8-bit register and can only be written when SE register is set to 0.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0xFC
- **Reset Value:** 0x00000000

Table 8-254 XIP Mode Bits Register

Bits	Field Name	RW	Reset	Description
15:0	MODE_BITS	RW		XIP mode bits to be sent after address phase of XIP transfer

8.5.2.4.27 XIP_INCR_INST

- **Name:** XIP INCR Transfer Opcode Register

- **Description:** This register is used to store the instruction op-code to be used in INCR transactions when the same is requested on AHB interface. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x100
- **Reset Value:** 0x00000000

Table 8-255 XIP INCR Transfer Opcode Register

Bits	Field Name	RW	Reset	Description
15:0	XIP_INCR_INST	RW	0x00	XIP INCR transfer opcode. When XIP_CTRL.INST_EN bit is set to 1, QSPI sends instruction for all XIP transfers, this register field stores the instruction op-code to be sent when an INCR type transfer is requested on AHB bus. The number of bits to be sent in instruction phase is determined by XIP_CTRL.INST_L field.

8.5.2.4.28 XIP_WRAP_INST

- **Name:** XIP WRAP Transfer Opcode Register
- **Description:** This register is used to store the instruction op-code to be used in WRAP transactions when the same is requested on AHB interface. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x104
- **Reset Value:** 0x00000000

Table 8-256 XIP WRAP Transfer Opcode Register

Bits	Field Name	RW	Reset	Description
15:0	XIP_WRAP_INST	RW	0x00	XIP WRAP transfer opcode. When XIP_CTRL.INST_EN bit is set to 1, QSPI sends instruction for all XIP transfers, this register field stores the instruction op-code to be sent when a WRAP type transfer is requested on AHB bus. The number of bits to be sent in instruction phase is determined by XIP_CTRL.INST_L field.

8.5.2.4.29 XIP_CTRL

- **Name:** XIP Control Register
- **Description:** This register is used to store the control information for the XIP transfer. It is impossible to write to this register when the QSPI is enabled.

- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x108
- **Reset Value:** 0x00000000

Table 8-257 XIP Control Register

Bits	Field Name	RW	Reset	Description
31:30	RSVD	R		Reserved bits
29	XIP_PREFETCH_EN	RW	0x0	Enables XIP pre-fetch functionality in QSPI. Once the functionality is enabled, QSPI will pre-fetch data frames from next contiguous location, to reduce the latency for the upcoming contiguous transfer. If the next XIP request is not contiguous, then pre-fetched bits will be discarded.
28	RSVD	R		Reserved bit
27:26	XIP_MBL	RW	0x0	XIP Mode bits length. Sets the length of mode bits in XIP mode of operation. These bits are valid only when XIP_CTRL.MD_BITS_EN is set to 1. **Value:** • 0x0 : Mode bits length equal to 2 • 0x1 : Mode bits length equal to 4 • 0x2 : Mode bits length equal to 8 • 0x3 : Mode bits length equal to 16
25:24	RSVD	R		Reserved bits
23	CONT_XFER_EN	RW	0x0	Enable continuous transfer in XIP mode. If this bit is set to 1, then continuous transfer mode in XIP will be enabled, in this mode QSPI will keep slave selected until a non-XIP transfer is detected on the AHB interface.
22	INST_EN	RW	0x0	XIP instruction enable bit. If this bit is set to 1, then XIP transfers will also have instruction phase. The instruction op-codes will be chosen from XIP_INCR_INST or XIP_WRAP_INST register based on AHB transfer type.
21:19	RSVD	R		Reserved bits
18	DFS_HC	RW	0x0	Fix DFS for XIP transfers. If this bit is set to 1, then data frame size for XIP transfers will be fixed to the programmed value in CTRL0.DFS. The number of data frames to fetch will be determined by HSIZE and HBURST signals. If this bit is

Bits	Field Name	RW	Reset	Description
				set to 0 then data frame size and number of data frames to fetch will be determined by HSIZE and HBURST signals.
17:13	WAIT_CYCLES	RW	0x0	Wait cycles in Dual/Quad mode between control frames transmit and data reception. Specified as number of QSPI clock cycles.
12	MD_BITS_EN	RW	0x0	Mode bits enable in XIP mode. If this bit is set to 1, then in XIP mode of operation, QSPI will insert mode bits after the address phase. These bits are set in XIP_MODE_BITS register. The length of mode bits is always set to 8 bits.
11	RSVD	R		Reserved bit
10:9	INST_L	RW	0x0	Dual/Quad mode instruction length in bits. **Value:** • 0x0 (INST_L0): No instruction • 0x1 (INST_L4): 4-bit instruction length • 0x2 (INST_L8): 8-bit instruction length • 0x3 (INST_L16): 16-bit instruction length
8	RSVD	R		Reserved bit
7:4	ADDR_L	RW	0x0	These bits define the length of address to be transmitted. Only after these bits are programmed into the FIFO, the transfer can begin. **Value:** • 0x0 (ADDR_L0): No address • 0x1 (ADDR_L4): 4-bit address length • 0x2 (ADDR_L8): 8-bit address length • 0x3 (ADDR_L12): 12-bit address length • 0x4 (ADDR_L16): 16-bit address length • 0x5 (ADDR_L20): 20-bit address length • 0x6 (ADDR_L24): 24-bit address length • 0x7 (ADDR_L28): 28-bit address length • 0x8 (ADDR_L32): 32-bit address length
3:2	TRANS_TYPE	RW	0x0	Address and instruction transfer format. Selects whether QSPI will transmit instruction/address either in standard SPI mode or the SPI mode selected in CTRL0.SPI_FRF field.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0: Instruction and address will be sent in standard SPI Mode. • 0x1: Instruction will be sent in standard SPI mode and address will be sent in the mode specified by CTRL0.SPI_FRF. • 0x2: Both instruction and address will be sent in the mode specified by SPI_FRF. • 0x3: Reserved.
1:0	FRF	RW	0x0	QSPI frame format. Selects data frame format for transmitting/receiving the data. **Value:** • 0x0 (RSVD): Reserved • 0x1 (SPI_DUAL): Dual SPI format • 0x2 (SPI_QUAD): Quad SPI format • 0x3 (RSVD): Reserved

8.5.2.4.30 XIP_SER

- **Name:** XIP Slave Enable Register
- **Description:** The register enables the individual slave select output lines from the QSPI master for XIP mode of operation. Users cannot write to this register when QSPI is busy or when QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x10C
- **Reset Value:** 0x00000000

Table 8-258 XIP Slave Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	SLAVE0	RW	0x0	Slave select enable flag

8.5.2.4.31 XIP_CNT_TIME_OUT

- **Name:** XIP Time Out Register for Continuous Transfers
- **Description:** XIP count down register for continuous mode. The counter is used to de-select the slave during continuous transfer mode. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000

- **Offset:** 0x114
- **Reset Value:** 0x00000000

Table 8-259 XIP Time Out Counter Register for Continuous Transfers

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7:0	XTOC	RW	0	XIP time out value in terms of HCLK. Once slave is selected in continuous XIP mode, this counter will be used to de-select the slave if there is no request for the time specified in the counter.

8.5.2.4.32 XIP_WR_INCR_INST

- **Name:** XIP Write INCR Transfer Opcode Register
- **Description:** This register is used to store the instruction op-code to be used in INCR transactions for XIP Write when the same is requested on AHB interface. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x140
- **Reset Value:** 0x00000000

Table 8-260 XIP Write INCR Transfer Opcode Register

Bits	Field Name	RW	Reset	Description
15:0	XIP_INCR_INST	RW	0x00	XIP Write INCR transfer opcode. When XIP_WR_CTRL.INST_L is not equal to 0, QSPI sends instruction for all XIP write transfers, this register field stores the instruction op-code to be sent when an INCR type XIP Write transfer is requested on AHB bus. The number of bits to be sent in instruction phase is determined by XIP_WR_CTRL.INST_L field.

8.5.2.4.33 XIP_WR_WRAP_INST

- **Name:** XIP Write WRAP Transfer Opcode Register
- **Description:** This register is used to store the instruction op-code to be used in WRAP transactions for XIP Write when the same is requested on AHB interface. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x144

- **Reset Value:** 0x00000000

Table 8-261 XIP Write WRAP Transfer Opcode Register

Bits	Field Name	RW	Reset	Description
15:0	XIP_WRAP_INST	RW	0x00	XIP Write WRAP transfer opcode. When XIP_WR_CTRL.INST_L is not equal to 0, QSPI sends instruction for all XIP write transfers, this register field stores the instruction op-code to be sent when a WRAP type XIP Write transfer is requested on AHB bus. The number of bits to be sent in instruction phase is determined by XIP_WR_CTRL.INST_L field.

8.5.2.4.34 XIP_WR_CTRL

- **Name:** XIP Write Control Register
- **Description:** This register is used to store the control information that will be used for XIP write transfer in the xip mode. It is impossible to write to this register when the QSPI is enabled.
- **Base Address:** QSPI0: 0x40023000; QSPI1: 0x40022000; QSPI2: 0x40021000
- **Offset:** 0x148
- **Reset Value:** 0x00000000

Table 8-262 XIP Write Control Register

Bits	Field Name	RW	Reset	Description
31:21	RSVD	R		Reserved bits
20:16	WR_CTRL_WAITCYCLES	RW	0x0	Wait cycles in Dual/Quad mode between control frames transmit and data reception. Specified as number of QSPI clock cycles.
15:10	RSVD	R		Reserved bits
9:8	WR_CTRL_INSTL	RW	0x0	Dual/Quad mode instruction length in bits. **Value:** • 0x0 (INST_L0): No Instruction • 0x1 (INST_L4): 4-bit Instruction length • 0x2 (INST_L8): 8-bit Instruction length • 0x3 (INST_L16): 16-bit Instruction length
7:4	WR_CTRL_ADDRL	RW	0x0	This bit defines Length of Address to be transmitted. Only after these bits are programmed into the FIFO, the transfer can begin.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0 (ADDR_L0): No Address • 0x1 (ADDR_L4): 4-bit Address length • 0x2 (ADDR_L8): 8-bit Address length • 0x3 (ADDR_L12): 12-bit Address length • 0x4 (ADDR_L16): 16-bit Address length • 0x5 (ADDR_L20): 20-bit Address length • 0x6 (ADDR_L24): 24-bit Address length • 0x7 (ADDR_L28): 28-bit Address length • 0x8 (ADDR_L32): 32-bit Address length • 0x9 (ADDR_L36): 36-bit Address length • 0xa (ADDR_L40): 40-bit Address length • 0xb (ADDR_L44): 44-bit Address length • 0xc (ADDR_L48): 48-bit Address length • 0xd (ADDR_L52): 52-bit Address length • 0xe (ADDR_L56): 56-bit Address length • 0xf (ADDR_L60): 60-bit Address length
3:2	WR_CTRL_TRANSTYPE	RW	0x0	Address and instruction transfer format. Selects whether QSPI will transmit instruction/address either in Standard SPI mode or the SPI mode selected in CTRL0.SPI_FRF field. **Value:** • 0x0: Instruction and Address will be sent in Standard SPI Mode. • 0x1: Instruction will be sent in Standard SPI Mode and Address will be sent in the mode specified by CTRL0.SPI_FRF. • 0x2: Both Instruction and Address will be sent in the mode specified by SPI_FRF. • 0x3: Reserved.
1:0	WR_CTRL_FRF	RW	0x0	QSPI frame format. Selects data frame format for Transmitting/Receiving the data. **Value:** • 0x0 (RSVD): Reserved • 0x1 (SPI_DUAL): Dual SPI Format

Bits	Field Name	RW	Reset	Description
				• 0x2 (SPI_QUAD): Quad SPI Format • 0x3 (RSVD): Reserved

8.5.2.4.35 QSPI_MODE_XIP

- **Name:** XIP Endian Mode Configure Register
- **Description:** This register is used to configure endian mode under XIP read access.
- **Base Address:** 0xA000E000
- **Offset:** 0x180
- **Reset Value:** 0x00000000

Table 8-263 XIP Endian Mode Configure Register

Bits	Field Name	RW	Reset	Description
31	M2_DYNAMIC_LE_EN	RW	0x0	Configure QSPI2 Dynamic Endian Mode. **Value:** • 0: Read and write 32-bit data without re-ordering in XIP mode • 1: Re-order the read and write data as [7:0], [15:8], [23:16], [31:24] when HSIZE is word. Re-order the read and write data as [23:16], [31:24], [7:0], [15:8] when HSIZE is half-word.
30	M1_DYNAMIC_LE_EN	RW	0x0	Configure QSPI1 Dynamic Endian Mode. **Value:** • 0: Read and write 32-bit data without re-ordering in XIP mode • 1: Re-order the read and write data as [7:0], [15:8], [23:16], [31:24] when HSIZE is word. Re-order the read and write data as [23:16], [31:24], [7:0], [15:8] when HSIZE is half-word.
29	M0_DYNAMIC_LE_EN	RW	0x0	Configure QSPI0 Dynamic Endian Mode. **Value:** • 0: Read and write 32-bit data without re-ordering in XIP mode • 1: Re-order the read and write data as [7:0], [15:8], [23:16], [31:24] when HSIZE is word. Re-order the read and write data as [23:16], [31:24], [7:0], [15:8] when HSIZE is half-word.
28:12	RSVD	RW	0x0	Reserved bits

Bits	Field Name	RW	Reset	Description
11:10	M2_STATIC_ENDIAN_MODE	RW	0x0	Configure QSPI2's static endian mode for XIP read. **Value:** • 0x0/0x3: Endian mode 0 • 0x1: Endian mode 1 • 0x2: Endian mode 2
9:8	M2_XIP_EN	RW	0x0	Control the QSPI2's XIP mode to turn to active or inactive. **Value:** • 0x0/0x2/0x3: active • 0x1: inactive
7:6	M1_STATIC_ENDIAN_MODE	RW	0x0	Configure QSPI1's static endian mode for XIP read. **Value:** • 0x0/0x3: Endian mode 0 • 0x1: Endian mode 1 • 0x2: Endian mode 2
5:4	M1_XIP_EN	RW	0x0	Control the QSPI1's XIP mode to turn to active or inactive. **Value:** • 0x0/0x2/0x3: active • 0x1: inactive
3:2	M0_STATIC_ENDIAN_MODE	RW	0x0	Configure QSPI0's static endian mode for XIP read. **Value:** • 0x0/0x3: Endian mode 0 • 0x1: Endian mode 1 • 0x2: Endian mode 2
1:0	M0_XIP_EN	RW	0x0	Control the QSPI0's XIP mode to be active or inactive. **Value:** • 0x0/0x2/0x3: active • 0x1: inactive

8.5.2.4.36 QSPI_CS_SETUP_DLY

- **Name:** CS Setup Delay Clocks Setting Register for QSPI
- **Description:** It works for both register and XIP access mode.
- **Base Address:** 0xA000E000
- **Offset:** 0x188

- **Reset Value:** 0x00000000

Table 8-264 CS Setup Delay Clocks Setting Register for QSPI

Bits	Field Name	RW	Reset	Description
31:24	RSVD	R	0x0	Reserved bits
23:16	M2_CS2SCK_DELAY	RW	0x0	Configure QSPI2's CS setup delay clocks. **Value:** • 0x0: • 0x1: … • 0x4:
15:8	M1_CS2SCK_DELAY	RW	0x0	Configure QSPI1's CS setup delay clocks. **Value:** • 0x0: • 0x1: … • 0x4:
7:0	M0_CS2SCK_DELAY	RW	0x0	Configure QSPI0's CS setup delay clocks. **Value:** • 0x0: • 0x1: … • 0x4:

8.5.2.4.37 QSPI_CS_RELEASE_DLY

- **Name:** CS Release Delay Clocks Setting Register for QSPI
- **Description:** It works for both register and XIP access mode.
- **Base Address:** 0xA000E000
- **Offset:** 0x18C
- **Reset Value:** 0x00000000

Table 8-265 CS Release Delay Clocks Setting Register for QSPI

Bits	Field Name	RW	Reset	Description
31:24	RSVD	R	0x0	Reserved bits

Bits	Field Name	RW	Reset	Description
23:16	M2_SCK2CS_DELAY	RW	0x0	Configure QSPI2's CS release delay clocks. **Value:** • 0x0: • 0x1: … • 0x4:
15:8	M1_SCK2CS_DELAY	RW	0x0	Configure QSPI1's CS release delay clocks. **Value:** • 0x0: • 0x1: … • 0x4:
7:0	M0_SCK2CS_DELAY	RW	0x0	Configure QSPI0's CS release delay clocks. **Value:** • 0x0: • 0x1: … • 0x4:

8.5.2.5 Electrical Specifications

Table 8-266 QSPI electrical specifications

Parameter	Description	Min.	Typ.	Max.	Unit
f_{sck}	Bit (baud) rate	47.96	48		MHz
t_{clk}	SCK clock period		20.83	20.85	ns
t_{rclk}	SCK clock rise time (10% -90%), 9pF Loading		1.45		ns
t_{fclk}	SCK clock fall time (90% - 10%), 9pF Loading		1.40		ns
t_{ch}	SCK clock high time		8.97		ns
t_{cl}	SCK clock low time		9.01		ns
t_{csp}	CS Setup delay time to active clock edge		10.4		ns
			20.8		ns

[5] *All values are measured under 48 MHz baud rate in Clock Mode 3.*

[6] *Default t_{csp} in XIP Mode and can be adjusted by either software or hardware.*

[7] *Default t_{csp} in Register Mode and can be adjusted by either software or hardware.*

Parameter	Description	Min.	Typ.	Max.	Unit
t_{sp}	Data Setup time to active clock edge			10.4	ns
				20.8	ns
t_{hd}	Data Hold time from active clock edge		8.97		ns

[5] All values are measured under 48 MHz baud rate in Clock Mode 3.
[6] Default t_{csp} in XIP Mode and can be adjusted by either software or hardware.
[7] Default t_{csp} in Register Mode and can be adjusted by either software or hardware.

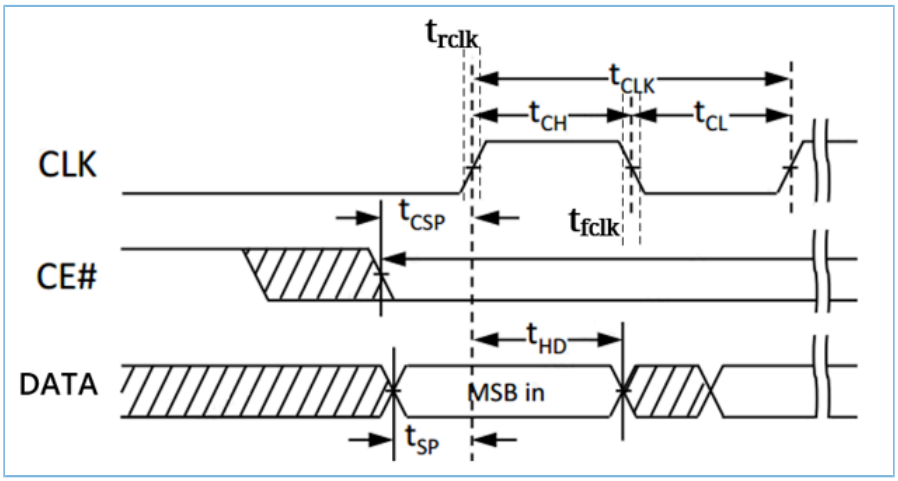

Figure 8-56 QSPI timing diagram

8.5.3 OSPI

8.5.3.1 Introduction

Octal Serial Peripheral Interface (OSPI) is an extended Serial Peripheral Interface (SPI) protocol. Compared with standard SPI, OSPI shares 8 data lines for transmitting and receiving. The communication mode is synchronous half-duplex. In the GR5526 SoC, the OSPI interface is only used to connect the PSRAM device supporting XccelaBus.

Note:

This module is only existed in graphics enhanced chips.

8.5.3.2 Main Features

Main features of the OSPI modules include:

- OSPI.DDR-PSRAM solution (Octal SPI with DDR Xccela mode, two bytes transferred per one clock cycle)
- 64 Mb 1.8V PSRAM (8 M x 8 bits with 1024 bytes page size)
- 48 MHz memory access frequency
- Variable latency and constant latency
- Adjustable drive strength

[5] *All values are measured under 48 MHz baud rate in Clock Mode 3.*

- Active mode, half-sleep mode, and deep power down
- Support for DMA (limited to aligned address)
- Support for global reset (using command instructions)

8.5.3.3 Functional Description

8.5.3.3.1 Block Diagram

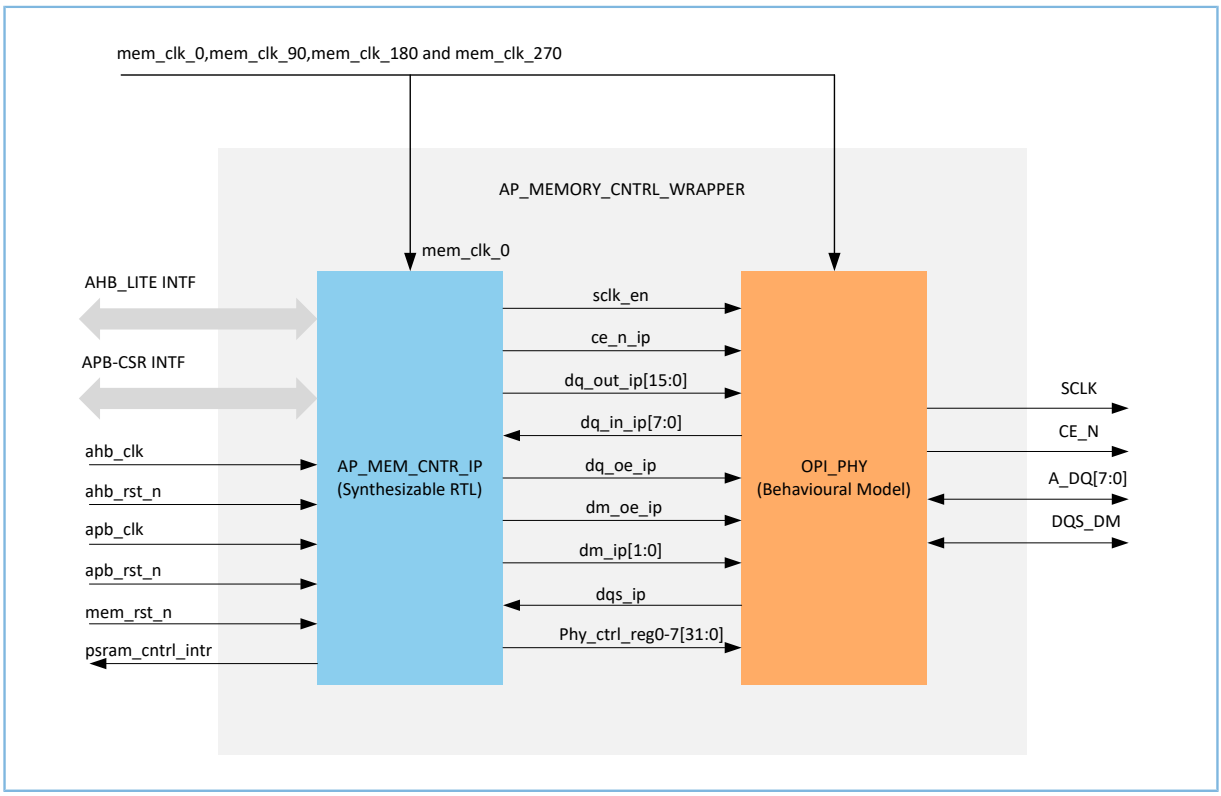

Figure 8-57 OSPI controller block diagram

The OSPI AHB-Lite memory controller has two system-side interfaces, namely APB and AHB-Lite interfaces.

The APB interface is used to access the control and status register (CSR) inside the controller. The AHB-Lite interface is used to write/read memory arrays and write/read mode registers in AP storage devices based on CSR register settings.

The AP_MEM_CNTR_IP instance is the digital controller IP. The OPI_PHY is responsible for generating AP memory interface signals and handling part of the timing maintenance together with the controller.

8.5.3.3.2 Control Interface

The commands specified on the INST (Instruction) cycle defined by the Address/DQ pins are latched on the first rising edge of the clock. The sampling address is latched on the rising/falling edge of the second or third clock. After address latching, the device initializes Data Strobe (DQS)/Data Mask (DM) to 0 from next CLK rising edge of the third clock cycle.

Output data is available after LC latency cycles. In case of internal refresh insertion, variable latency output data may be delayed by up to (LC*2) latency cycles. True variable refresh pushout latency can be anywhere between LC to LCx2. The first DQS/DM rising edge after reading the preamble indicates the beginning of valid data.

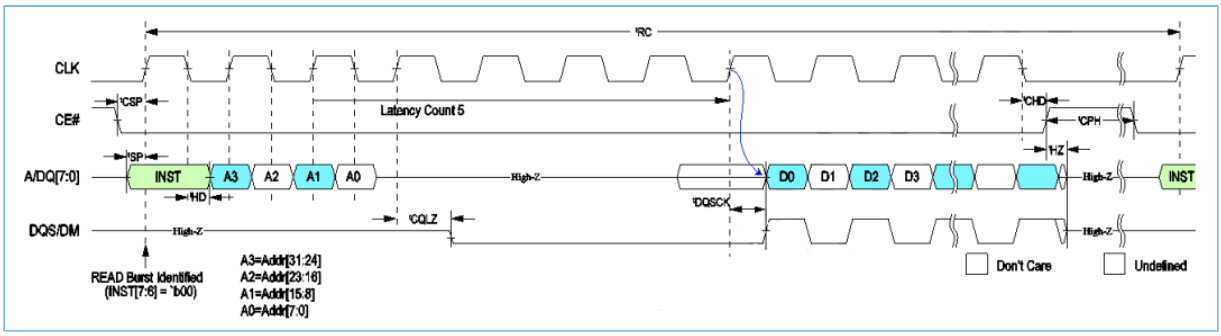

Figure 8-58 Synchronous read

Extra payload may exist in reading. Due to the complexity of Xccela control, the master logic cannot switch the state in time after reading the expected data, causing the slave to continue to generate unexpected data. Because this part of the data is not considered as the payload, it will reduce the access efficiency. Using MCU to access OSPI in reading will generate more extra DQS clocks than using DMA/GPU. Excessive overhead will reduce access efficiency. The access mode of MCU under Cortex®-M4F is INCR. Bursts will affect the access efficiency so that the practice of using MCU to access OSPI in reading should be minimized.

At least 2 bytes of data must be input in the write operation. In the case of continuous short burst writes, tRC must be met by issuing additional CE# high time between operations. Single-byte write operations can be performed by using DQS/DM to mask unwritten bytes.

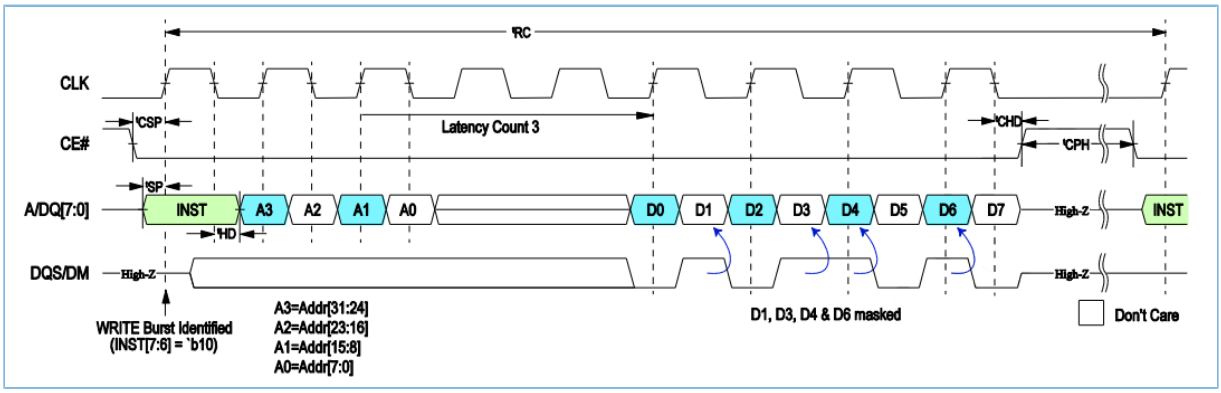

Figure 8-59 Synchronous write followed by any operation

It is worth noting that PSRAM has been embedded in GR5526, but it needs to be enabled before using OSPI controller. By default, the PSRAM is disabled considering the impact of power consumption of PSRAM. The impedance matching between OSPI controller and PSRAM can be modified by adjusting the drive strength of PSRAM. Lower drive strength indicates lower power consumption. The waveform tends to be triangle wave, and the quality of the waveform is worse. Greater drive strength indicates higher power consumption. The waveform tends to be square wave, and the quality of the waveform is better. However, excessive drive strength may lead to system crash. Therefore, an appropriate drive strength should be selected according to the application scenario.

8.5.3.4 Registers

8.5.3.4.1 MEM_BASE_ADDR

- **Name:** Bottom/Base Memory Address Register
- **Description:** This register represents the bottom/base address of the memory.
- **Base Address:** 0x40024000
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 8-267 Bottom/Base Memory Address Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:0	MEM_BASE_ADDR	RW	0x0	This represents the bottom/base address of the memory. **Note:** This reset value is for the bottom array of a memory.

8.5.3.4.2 MEM_TOP_ADDR

- **Name:** Top Memory Address Register
- **Description:** This register represents the top address of the memory.
- **Base Address:** 0x40024000
- **Offset:** 0x4
- **Reset Value:** 0x01FFFFFF

Table 8-268 Top Memory Address Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:0	MEM_TOP_ADDR	RW	0x1FFFFFF	This represents the top address of the memory. **Note:** This reset value is based on 256 Mb device density memory.

8.5.3.4.3 GLOBAL_RESET

- **Name:** Global Reset Register
- **Description:** This register configures global reset.
- **Base Address:** 0x40024000
- **Offset:** 0x8

- **Reset Value:** 0x00000216

Table 8-269 Global Reset Register

Bits	Field Name	RW	Reset	Description
31:10	RSVD	R		Reserved bits
9:1	TRST_CNT	RW	0x10B	This holds the CE# high time in terms of mem_clk cycles between reset command execution and subsequent write/read operation. **Note:** The reset value corresponds to 2 µs at 133 MHz.
0	GLOBAL_RESET_EN	RW	0x0	1: Enables the global reset instruction to be initiated to the memory. **Note:** This bit is cleared to zero by itself after this transaction completes with the memory. Users should not write zero to this bit.

8.5.3.4.4 ACCESS_TYPE

- **Name:** Access Type Register
- **Description:** This register controls access type.
- **Base Address:** 0x40024000
- **Offset:** 0xC
- **Reset Value:** 0x00000001

Table 8-270 Access Type Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	MR_ACCESS	RW	0x1	Memory array or mode register access 0: Memory array access 1: Mode register access

8.5.3.4.5 ACCESS_TIMING

- **Name:** Access Timing Register
- **Description:** This register sets TCEM, TRC, TCPH timing, and page size.
- **Base Address:** 0x40024000

- **Offset:** 0x10
- **Reset Value:** 0x00A50840

Table 8-271 Access Timing Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24	EN_EFF_THRGHPUT	RW	0x0	Enable efficient read throughput. 1: Enhance the read throughput by enabling this bit in CSR, TCEM_TRC_PAGE_SIZE_RD_THRGHPUT_REG [tcem_time_ignore]. This bit should be set to 1 and also the row boundary crossing in the PSRAM memory should be enabled.
23:20	MEM_PAGE_SIZE	RW	0xA	Page size in memory = 2^(mem_page_size) • 0x7: 128 bytes • 0x8: 256 bytes • 0x9: 512 bytes • 0xA: 1024 bytes
19:17	TCPH_CNT	RW	0x2	This holds the CE# HIGH time in terms of mem_clk cycles between subsequent burst operations in terms of mem_clk cycles. **Note:** The reset value corresponds to 15 ns at 133 MHz.
16:13	TRC_CNT	RW	0x8	This holds the write and read cycle time (tRC) in terms of mem_clk cycles. **Note:** The reset value corresponds to 60 ns at 133 MHz.
12:1	TCEM_CNT	RW	0x420	This is the CE# low pulse width time in terms of mem_clk cycles, during any memory array access. The default value is set based on 133 MHz memory clock operation.
0	TCEM_TIME_IGNORE	RW	0x0	1: If tCEM time needs not be taken care by the controller, then F/W shall set this bit. **Note:** This bit should be set to 1 during WRAP transfers.

8.5.3.4.6 DEEP_DOWN_CTRL

- **Name:** Deep Down Control Register
- **Description:** This register controls deep power down.
- **Base Address:** 0x40024000
- **Offset:** 0x14
- **Reset Value:** 0x00427100

Table 8-272 Deep Down Control Register

Bits	Field Name	RW	Reset	Description
31:23	RSVD	R		Reserved bits
22:19	DPD_EXIT_CYCLE_CNT	RW	0x8	This decides the tXPDPD time during deep power down exit. In multiples of mem_clk cycles (Min value of 60 ns). The default value is set based on 133 MHz memory clock operation and tXPDPD time 60 ns.
18:3	TXDPD_COUNT	RW	0x4E20	This decides the tXDPD time during deep power down exit in terms of mem_clk cycles. The default value is set based on 133MHz memory clock operation and tXDPD time 150 µs.
2	TXDPD_TIME_IGNORE	RW	0x0	1: If F/W can take care of the tXDPD time, it can set this bit. By setting this bit, the controller shall not run an internal counter (based on mem_clk) to take care of the tXDPD time. Hence, the system can shut off the mem_clk to the controller.
1	DPD_EXIT	RW	0x0	1: Triggers the deep power down exit instruction to be sent to the memory. **Note:** This bit is cleared to zero by itself after this transaction completes with the memory. Users should not write zero to this bit.
0	DPD_ENTRY	RW	0x0	1: Triggers the deep power down entry instruction to be sent to the memory. **Note:** This bit is cleared to zero by itself after this transaction completes with the memory. Users should not write zero to this bit.

8.5.3.4.7 HALF_SLP_CTRL

- **Name:** Half Sleep Control Register
- **Description:** This register controls half sleep.
- **Base Address:** 0x40024000
- **Offset:** 0x18
- **Reset Value:** 0x0041A0B0

Table 8-273 Half Sleep Control Register

Bits	Field Name	RW	Reset	Description
31:22	RSVD	R		Reserved bits
21:18	HS_EXIT_CYCLE_CNT	RW	0x8	This decides the tXPHS time during half-sleep exit. In multiples of mem_clk cycles (min value of 60 ns). The default value is set based on 133 MHz memory clock operation and tXPHS time 60 ns
17:3	TXHS_COUNT	RW	0x3416	This decides the tXHS time during half sleep exit in terms of mem_clk cycles. The default value is set based on 133 MHz memory clock operation and tXHS time 100 µs.
2	TXHS_TIME_IGNORE	RW	0x0	1: If F/W can take care of the tXHS time, it can set this bit. By setting this bit, the controller shall not run an internal counter (based on mem_clk) to take care of the tXHS time. Hence, the system can shut off the mem_clk to the controller.
1	HS_EXIT	RW	0x0	1: Triggers the Half sleep exit instruction to be sent to the memory. **Note:** This bit is cleared to zero by itself after this transaction completes with the memory. Users should not write zero to this bit.
0	HS_ENTRY	RW	0x0	1: Triggers the Half sleep entry instruction to be sent to the memory. **Note:** This bit is cleared to zero by itself after this transaction completes with the memory. Users should not write zero to this bit.

8.5.3.4.8 INT_CTRL

- **Name:** Interrupt Control Register
- **Description:** This register controls interrupt kinds.
- **Base Address:** 0x40024000
- **Offset:** 0x1C
- **Reset Value:** 0x00000000

Table 8-274 Interrupt Control Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	DQS_TIMEOUT_IE	RW	0x0	1: Generates an interrupt when DQS timeout occurs during read operation.
4	DPD_EXIT_IE	RW	0x0	1: Generates an interrupt when deep power down exit transfer to the AP memory device completes.
3	DPD_ENTRY_IE	RW	0x0	1: Generates an interrupt when deep power down entry transfer to the AP memory device completes.
2	HS_EXIT_IE	RW	0x0	1: Generates an interrupt when half-sleep exit transfer to the AP memory device completes.
1	HS_ENTRY_IE	RW	0x0	1: Generates an interrupt when half-sleep entry transfer to the AP memory device completes.
0	GLOBAL_RESET_IE	RW	0x0	1: Generates an interrupt when global reset transfer to the AP memory device completes.

8.5.3.4.9 XFER_STAT

- **Name:** Transfer Status Register
- **Description:** This register shows transfer status.
- **Base Address:** 0x40024000
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 8-275 Transfer Status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	DQS_NON_TOGGLE_ERR	RC	0x0	This bit is set if “DQS” from the memory is not toggling during AHB-Lite initiated memory read/mode register read operation.

Bits	Field Name	RW	Reset	Description
4	DPD_EXIT_DONE	RC	0x0	This bit is set to denote that the APB-CSR initiated deep power down exit transfer completes.
3	DPD_ENTRY_DONE	RC	0x0	This bit is set to denote that the APB-CSR initiated deep power down entry transfer completes.
2	HS_EXIT_DONE	RC	0x0	This bit is set to denote that the APB-CSR initiated half sleep exit transfer completes.
1	HS_ENTRY_DONE	RC	0x0	This bit is set to denote that the APB-CSR initiated entry transfer completes.
0	GLOBAL_RESET_DONE	RC	0x0	This bit is set to denote that the APB-CSR initiated global reset transfer completes.

8.5.3.4.10 CMD_CTRL_1

- **Name:** Command Control Register 1
- **Description:** This register is command control register.
- **Base Address:** 0x40024000
- **Offset:** 0x24
- **Reset Value:** 0xA0208000

Table 8-276 Command Control Register 1

Bits	Field Name	RW	Reset	Description
31:24	LINEAR_BURST_WR	RW	0xA0	This holds the command to be sent to the memory during linear burst write operation.
23:16	LINEAR_BURST_RD	RW	0x20	This holds the command to be sent to the memory during linear burst read operation.
15:8	SYNC_WR	RW	0x80	This holds the command to be sent to the memory during sync write operation
7:0	SYNC_RD	RW	0x0	This holds the command to be sent to the memory sync read operation.

8.5.3.4.11 CMD_CTRL_2

- **Name:** Command Control Register 2
- **Description:** This register is command control register.
- **Base Address:** 0x40024000
- **Offset:** 0x28

- **Reset Value:** 0x00FFC040

Table 8-277 Command Control Register 2

Bits	Field Name	RW	Reset	Description
31:24	RSVD	R		Reserved bits
23:16	GLOBAL_RESET_CMD	RW	0xFF	This holds the command to be sent to the memory during global reset operation.
15:8	MODE_REG_WR	RW	0xC0	This holds the command to be sent to the memory during mode register write operation.
7:0	MODE_REG_RD	RW	0x40	This holds the command to be sent to the memory during mode register read operation.

8.5.3.4.12 DQS_TIMEOUT

- **Name:** DQS Timeout Register
- **Description:** This register sets DQS timeout.
- **Base Address:** 0x40024000
- **Offset:** 0x2C
- **Reset Value:** 0x00000000

Table 8-278 DQS Timeout Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4:0	DQS_NON_TGL_TIMEOUT	RW	0x0	Read DQS non-toggling timeout. This is to set the tRBXwait (as per spec.) This holds the maximum number of mem_clk cycles for which the controller has to wait for the DQS toggling during row boundary crossing in a read operation.

8.5.3.4.13 PHY_CTRL_0

- **Name:** PHY Control Register 0
- **Description:** This register controls PHY.
- **Base Address:** 0x40024000
- **Offset:** 0x30
- **Reset Value:** 0x00000000

Table 8-279 PHY Control Register 0

Bits	Field Name	RW	Reset	Description
31:0	PHY_CNTRL_REG0	RW	0x0	PHY control register 0

8.5.3.4.14 PHY_CTRL_1

- **Name:** PHY Control Register 1
- **Description:** This register controls PHY.
- **Base Address:** 0x40024000
- **Offset:** 0x34
- **Reset Value:** 0x00000000

Table 8-280 PHY Control Register 1

Bits	Field Name	RW	Reset	Description
31:0	PHY_CNTRL_REG1	RW	0x0	PHY control register 1

8.5.3.4.15 PHY_CTRL_2

- **Name:** PHY Control Register 2
- **Description:** This register controls PHY.
- **Base Address:** 0x40024000
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 8-281 PHY Control Register 2

Bits	Field Name	RW	Reset	Description
31:0	PHY_CNTRL_REG2	RW	0x0	PHY control register 2

8.5.3.4.16 PHY_CTRL_3

- **Name:** PHY Control Register 3
- **Description:** This register controls PHY.
- **Base Address:** 0x40024000
- **Offset:** 0x3C
- **Reset Value:** 0x00000000

Table 8-282 PHY Control Register 3

Bits	Field Name	RW	Reset	Description
31:0	PHY_CNTRL_REG3	RW	0x0	PHY control register 3

8.5.3.4.17 PHY_CTRL_4

- **Name:** PHY Control Register 4
- **Description:** This register controls PHY.
- **Base Address:** 0x40024000
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 8-283 PHY Control Register 4

Bits	Field Name	RW	Reset	Description
31:0	PHY_CNTRL_REG4	RW	0x0	PHY control register 4

8.5.3.4.18 PHY_CTRL_5

- **Name:** PHY Control Register 5
- **Description:** This register controls PHY.
- **Base Address:** 0x40024000
- **Offset:** 0x44
- **Reset Value:** 0x00000000

Table 8-284 PHY Control Register 5

Bits	Field Name	RW	Reset	Description
31:0	PHY_CNTRL_REG5	RW	0x0	PHY control register 5

8.5.3.4.19 PHY_CTRL_6

- **Name:** PHY Control Register 6
- **Description:** This register controls PHY.
- **Base Address:** 0x40024000
- **Offset:** 0x48
- **Reset Value:** 0x00000000

Table 8-285 PHY Control Register 6

Bits	Field Name	RW	Reset	Description
31:0	PHY_CNTRL_REG6	RW	0x0	PHY control register 6

8.5.3.4.20 PHY_CTRL_7

- **Name:** PHY Control Register 7
- **Description:** This register controls PHY.
- **Base Address:** 0x40024000
- **Offset:** 0x4C
- **Reset Value:** 0x00000000

Table 8-286 PHY Control Register 7

Bits	Field Name	RW	Reset	Description
31:0	PHY_CNTRL_REG7	RW	0x0	PHY control register 7

8.5.3.4.21 RD_PREFETCH

- **Name:** Read Prefetch Register
- **Description:** This register controls read prefetch.
- **Base Address:** 0x40024000
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 8-287 Read Prefetch Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RD_PREFETCH_EN	RW	0x0	1: Read data pre-fetch is enabled in the controller for read operation. 0: Read data prefetch is disabled in the controller for read operation.

8.5.4 DC

8.5.4.1 Introduction

The display control (DC) module is a powerful display controller which contains several smart tools and functionalities to compose multiple graphics layers by improving image quality and contributing significantly to the reduction of

the SoC power consumption. The DC module supports composition features, a wide range of display interfaces, and advanced proprietary framebuffer compression technology. The core is designed to lift the workload off the Graphics Processing Unit (GPU) or the host processor (CPU), in GPU-less systems, and minimize the memory bandwidth.

8.5.4.2 Main Features

- 1 instance
- Supporting up to two layers with blending capabilities
- Programmable display resolutions up to 454 x 454
- Compressed framebuffer support
- Powerful composition
- Alpha blending
- Programmable size, offset, and format per layer
- Programmable stride/pitch enabling panning and clipping
- SPI3/SPI4/DualSPI/QuadSPI display interface
- DMA support (internal dedicated DMA)
- Input color formats: GBA-8888, ARGB-8888, ABGR-8888, BGRA-8888, RGB-888, RGBA-5551, RGB-565, RGB-332, RGB-565, TSC4,TSC6, TSC6a
    - 3/4w SPI: RGB-111, RGB-332, RGB-444, RGB-565, RGB-666, RGB-888
    - Dual/Quad SPI: RGB-444, RGB-565, RGB-666, RGB-888

8.5.4.3 Block Diagram

The functional block diagram of display control is shown in Figure 8-60. The display control module contains timing generator, master arbiter, two layers, formatting block, and more.

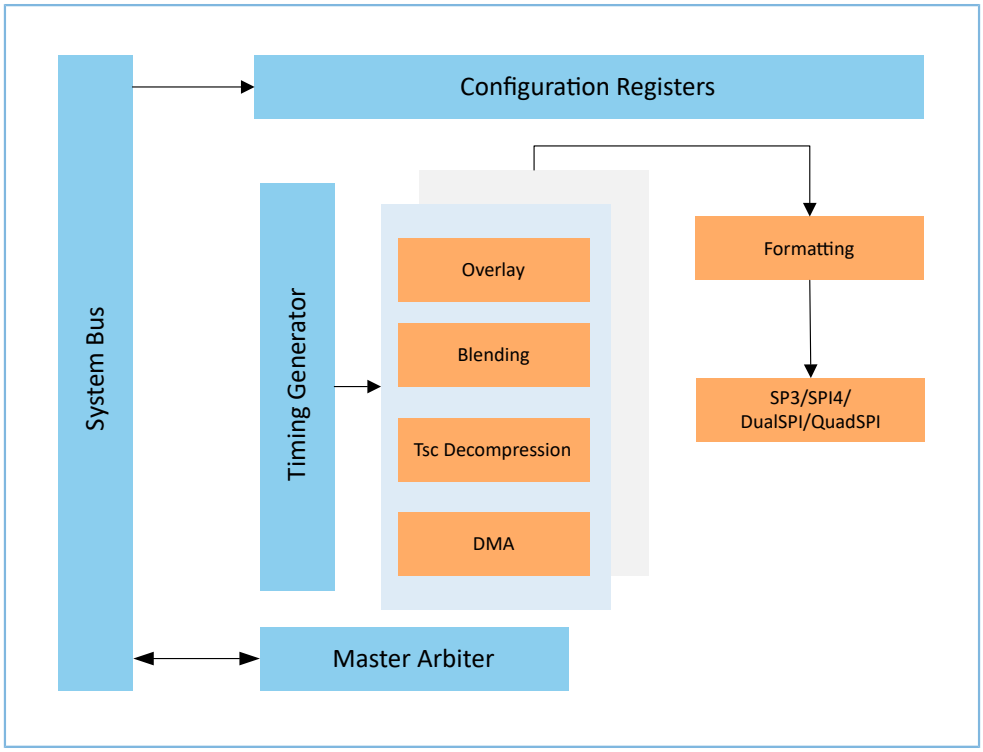

Figure 8-60 Display control block diagram

8.5.4.4 Functional Description

8.5.4.4.1 Pin Mux

Before configuring the display control module functions, you must ensure that the pin mux of the display control interface GPIOs should be configured as DC_DCX, DC_CS_N, DC_CLK, DC_IO_0, DC_IO_1, DC_IO_2, and DC_IO_3.

8.5.4.4.2 Clock Divider

The input PLL clock (pll_clk) which is serial clock is fed to the clock divider module. If the optional divider is enabled, then its input is driven by the PLL clock and its ratio is determined by the CLK_CTRL[31:27] register bits. Primary clock's divider input is driven by the optional divider output, if it is enabled; otherwise by the PLL clock. Primary clock's divider ratio is determined by the CLK_CTRL[5:0] register bits. In case that SPI interface is selected, the SPI_SCLK is generated by the format_clk, and then a hard division between the SPI_SCLK and the format_clk is applied depending on the INTERFACE_CFG[10:9]. The format_clk and pixel_clk are also fed to Pixel Pipeline and Display Format, representing the different processing stages of the Display controller.

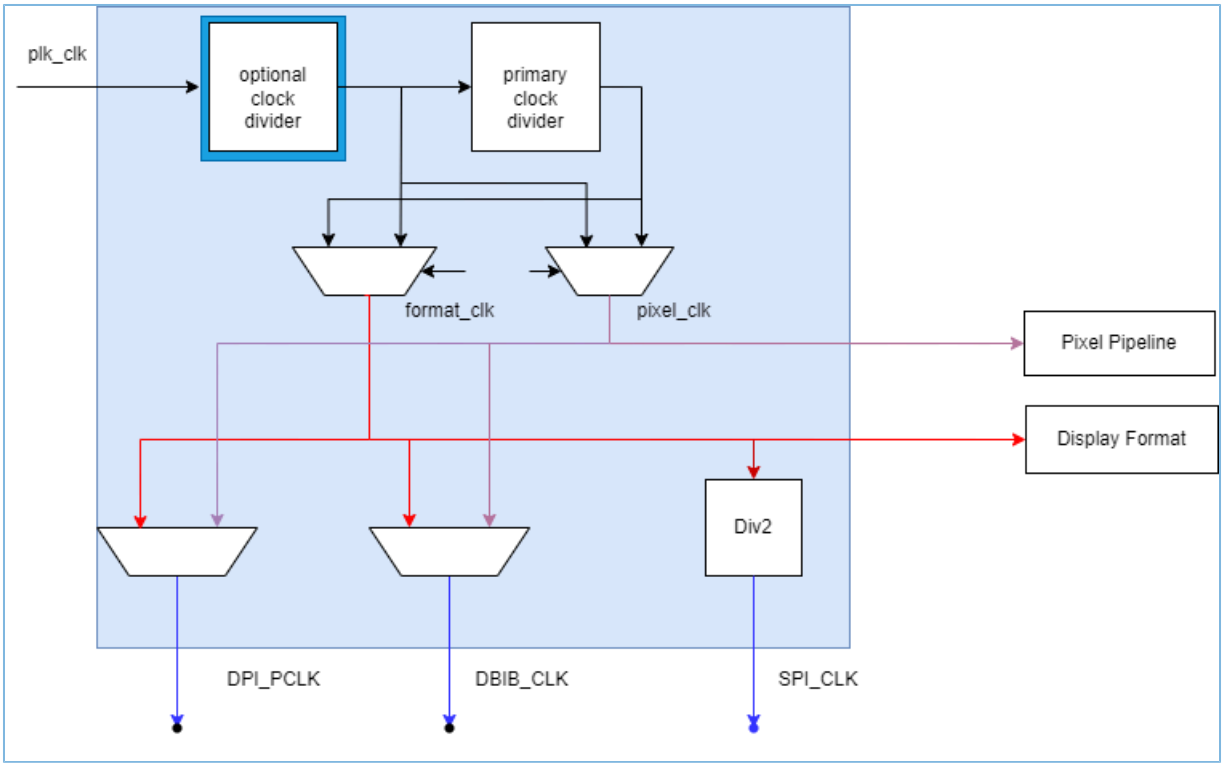

Figure 8-61 Clock division diagram

8.5.4.4.3 Layer Overlays

The DC module supports two layers (n = 0, 1) sourced from different memory regions. Each layer can have separate color modes, alpha blending, and filtering attributes. The main control registers for each layer are the LAYERn_MODE (n = 0–1) registers. Layer 0 is on top of the background color, and Layer 1 is on top of Layer 0.

Layering starts with a 32-bit background color (RGBA), as shown in Figure 8-62, which is applied on the entire screen. If it is not needed, the background register BG_COLOR field is set to 0. If no layer is enabled, only the background color will be displayed.

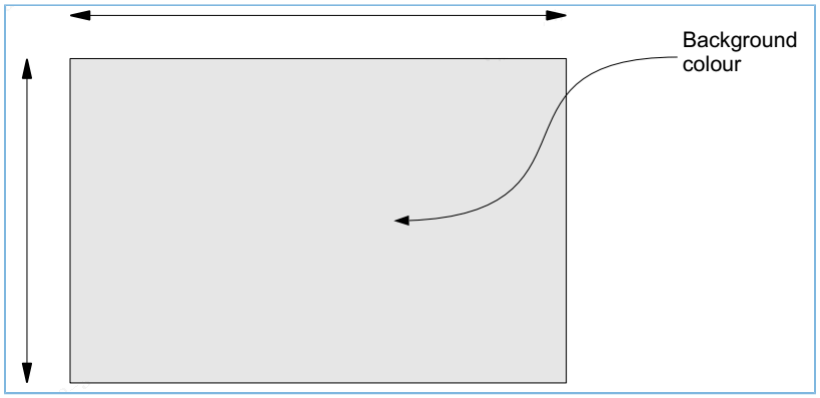

Figure 8-62 RGBA background color

The base frame address is set in the LAYERn_BASEADDR (n = 0–1) registers. When the video image is stored in the memory, the frame buffer may contain extra bytes after each row of pixels. The number of bytes from one row of

pixels to the next row in memory is called stride. These bytes affect how the image is stored in memory and the LAYERn_STRIDE[15:0] (n = 0–1) bits are used to add space between frame lines.

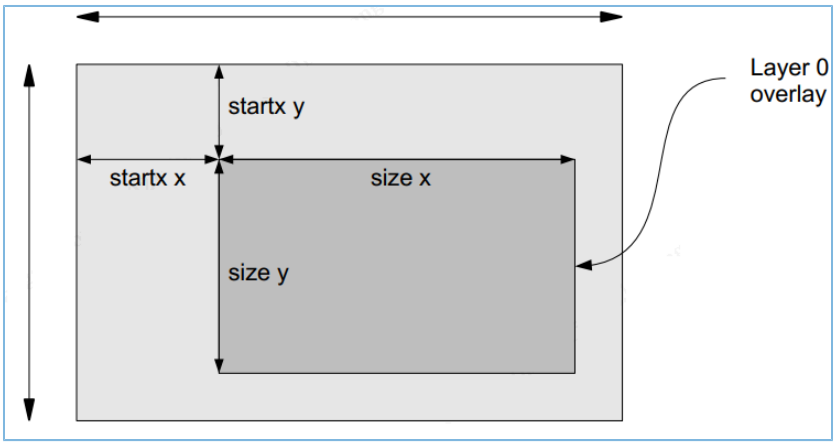

Figure 8-63 Layer overlay

For each layer, the start location (registers LAYERn_STARTXY n = 0–1), the visible size (registers LAYERn_SIZEXY n= 0–1), and the resolution (registers LAYERn_RESXY n = 0–1) are needed. For example, if the resolution of an image is 800 x 600 and the size of the required image for layer 0 is 400 x 300 at location (68, 32), then the value of the registers are shown in Table 8-288:

Table 8-288 Layer0 example

Parameter	Register	Register Bits	Value
Resolution X	RESXY	31–16	800
Resolution Y	RESXY	15–0	600
Start coordinate X	LAYER0_STARTXY	31–16	68
Start coordinate Y	LAYER0_STARTXY	15–0	32
Pixel size in X coordinate	LAYER0_SIZEXY	31–16	400
Pixel size in Y coordinate	LAYER0_SIZEXY	15–0	300

8.5.4.4.4 Blending Modes

Each layer supports alpha blending. During blending process, a translucent foreground color (current layer) with a background color (previous layer) are combined, and a new blended color is produced. Foreground color's translucency may range from completely transparent to completely opaque. If the foreground color is completely transparent, the blended color will be the background color; if the foreground color is completely opaque, the blended color will be the foreground color. When the translucency ranges in between, the blended color is computed as a weighted average of the foreground and background colors. Each layer is blended on top of the previous generated blended layer based on the following equation:

$$c = c_s \times F_s + c_d \times F_d \tag{23}$$

Table 8-289 shows the blending modes that the DC module supports:

Table 8-289 Blending modes

Binary	Blending Mode	F_s	F_d
0000	Blend black	0	0
0001	Blend white	1	1
0010	Blend alpha source	a_s	a_s
0011	Blend alpha global	a_g	a_g
0100	Blend alpha source and alpha global	$a_s * a_g$	$a_s * a_g$
0101	Blend inverted source	$1 – a_s$	$1 – a_s$
0110	Blend inverted global	$1 – a_g$	$1 – a_g$
0111	Blend inverted source and inverted global	$1 – (a_s * a_g)$	$1 – (a_s * a_g)$
1010	Blend alpha destination	a_d	a_d
1101	Blend inverted destination	$1 – a_d$	$1 – a_d$

Note:

a_s: The A channel of the source image, representing the transparency of the source image

a_s: The A channel of the destination image, representing the transparency of the destination image

a_g: The global A channel

8.5.4.4.5 TSC Format

The DC's framebuffer compression operates in screen blocks (4 x 4 pixel blocks), and depending on the configuration achieves TSC™4, TSC™6, and TSC™6a lossy, fixed-ratio compression.

- TSC™4 is a 6:1 compression (4bpp).
- TSC™6 is a 4:1 compression (6bpp).
- TSC™6a is a 4:1 compression (6bpp) with alpha channel.

Compression is performed at run time using minimal hardware. Pixel data can be stored in the framebuffer in compressed form and decompressed in the DC module. Figure 8-64 shows the TS compression operation. The output of the TSC™4 compression is 64 bits per 4 x 4 block of pixels, and the output of the TSC™6 compression is 96 bits per 4 x 4 block of pixels. It should be noted that the three compression formats of TSC are all lossy compression, and users need to measure whether the format meets their requirements for reduction.

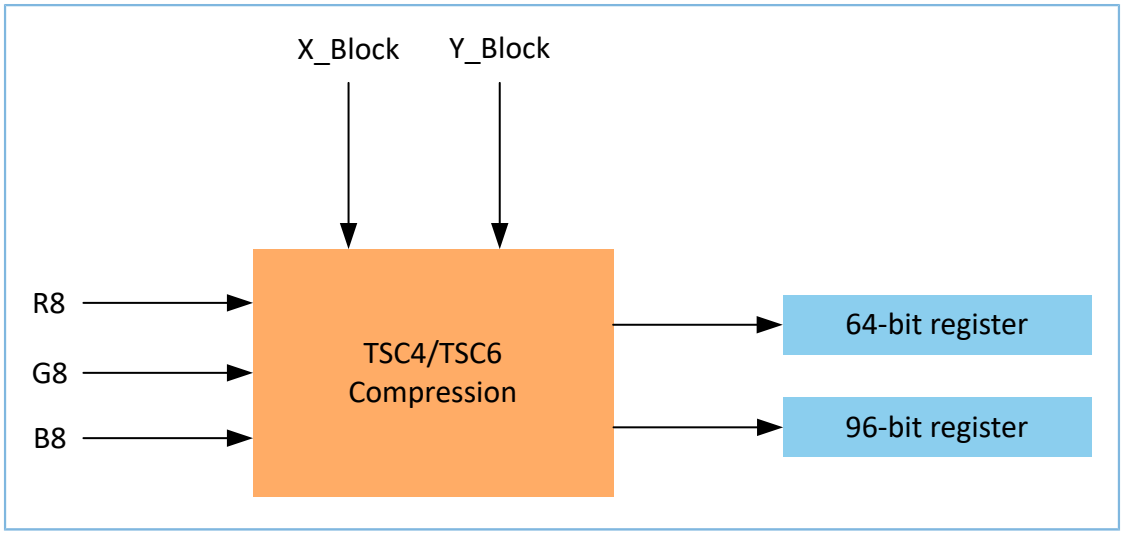

Figure 8-64 TSC™4 /TSC™6 framebuffer compression module

8.5.4.4.6 Display Interface

The DC module supports one-/multiple-line SPI. All supported formats are listed as follows:

- 3-wire SPI
- 4-wire SPI
- Dual SPI
- Quad SPI

The SPI serial interface is mainly composed of five parts:

- SPI_DC: data/command indicator
- SPI_CS: chip enable
- SPI_SCLK: serial clock
- SPI_SD/SPI_SD1/SPI_SD2/SPI_SD3: serial data output
- SPI_SI: serial data input

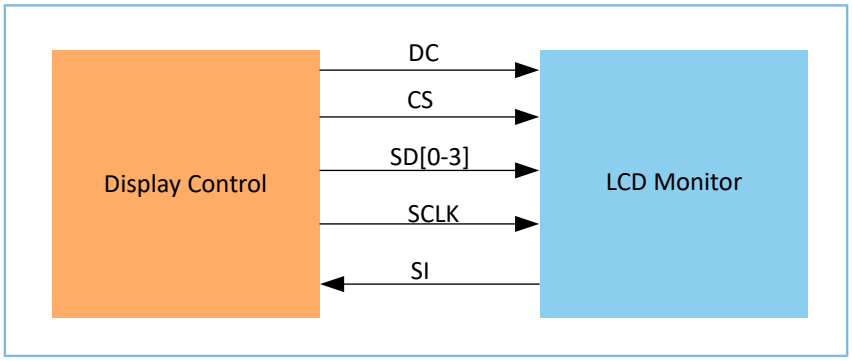

Figure 8-65 SPI interface

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

8.5.4.4.6.1 SPI 3-Wire/4-Wire Serial Interface

The SPI 3-wire serial interface uses the SPI_CS (chip enable), SPI_SCLK (serial clock), and SPI_SD (serial data output). The command word consists of 9 bits: 1-bit data/command indicator and 8-bit command. The data word consists of 1-bit data/command indicator and as many bits as required from the chosen format. The MSB is transmitted first.

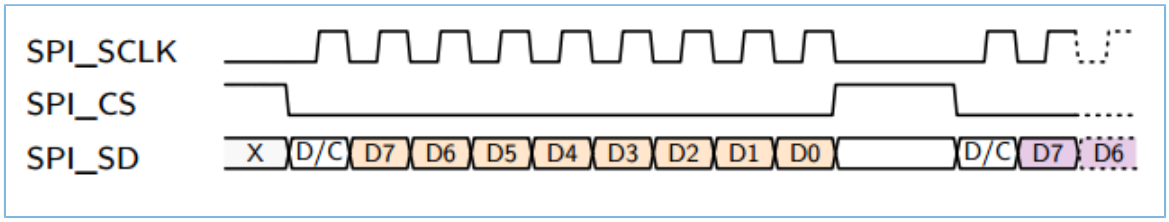

Figure 8-66 SPI 3-wire serial write transmission

The SPI 4-wire serial interface uses SPI_CS (chip enable), SPI_SCLK (serial clock), SPI_SD (serial data output), and one additional signal SPI_DC (data/command indicator). The command consists of 8 bits. The data consists of n bits, according to the chosen format. The MSB is transmitted first.

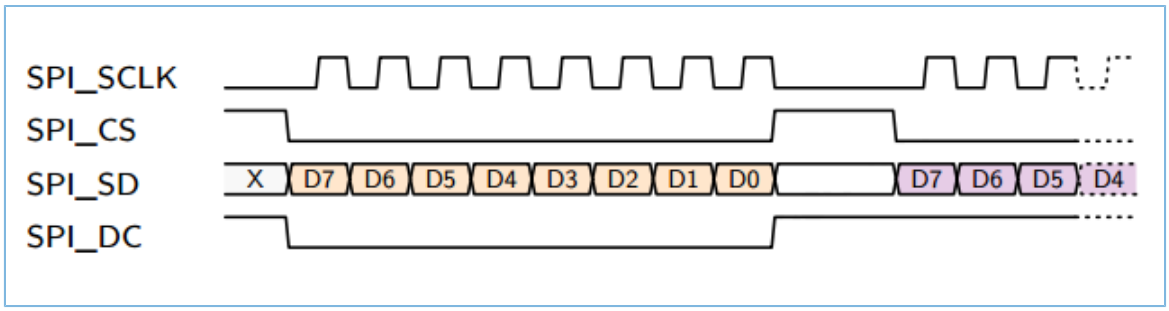

Figure 8-67 SPI 4-wire serial write transmission

The waveform of the read data is as follows:

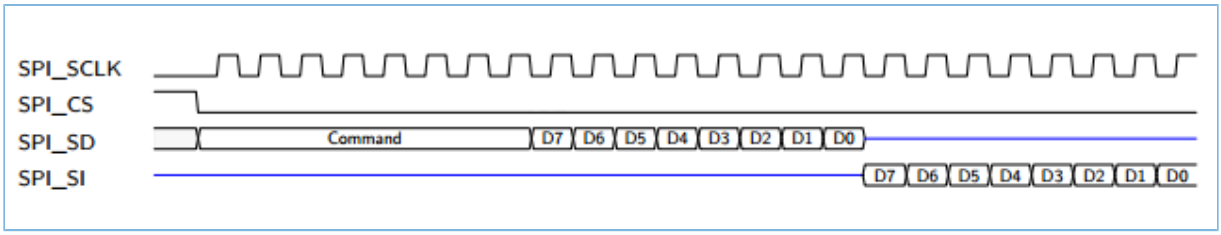

Figure 8-68 SPI read

8.5.4.4.6.2 DualSPI Serial Interface

The DualSPI serial interface uses SPI_CS (chip enable), SPI_SCLK (serial clock), and SPI_SD & SPI_SD1 (serial data outputs). The command word is always transmitted through the SPI_SD line, and the data is routed through the SPI_SD and SPI_SD1 lines. The command word consists of 9 bits: 1-bit data/command indicator and 8-bit command. The data word consists of 1-bit data/command indicator and n bits, according to the chosen format.

8.5.4.4.6.3 QuadSPI Serial Interface

The QuadSPI serial interface uses SPI_CS (chip enable), SPI_SCLK (serial clock), as well as SPI_SD, SPI_SD1, SPI_SD2, and SPI_SD3 (serial data outputs). The header and the command type are always transmitted through the SPI_SD line, and the data is routed through the SPI_SD, SPI_SD1, SPI_SD2, and SPI_SD3 lines. The header consists of 8 bits. The command word consists of 24 bits, and the data consists of n bits according to the chosen format.

The available signals of the SPI3/4, DualSPI, and QuadSPI interfaces are shown in the table below.

Table 8-290 Available signals of SPI3/4, DualSPI, and QuadSPI interfaces

Pin Name	Type	Description	Source
SCLK	Output	Serial clock signal	Display Controller
CS	Output	Chip enable when low	Display Controller
DC	Output	Data/Command select	Display Controller
SD0	Output	Serial data	Display Controller
SD1	Output	Serial data	Display Controller
SD2	Output	Serial data	Display Controller
SD3	Output	Serial data	Display Controller
SI	Input	Serial data (Read)	Display Controller
RESET	Output	Reset display	GPIO
TE	Input	Display Tearing effect signal	GPIO
Backlight	Output	LED driver signals	GPIO

8.5.4.4.7 Color Input Format

The DC module supports multiple color formats. The supported color formats are the following:

- L1 big-endian 1-bit

Table 8-291 L1 big-endian 1-bit

Pixel 0	Pixel 1	Pixel 2	Pixel 3	Pixel 4	Pixel 5	Pixel 6	Pixel 7
7	6	5	4	3	2	1	0
L0	L0	L0	L0	L0	L0	L0	L0
Byte 0							

- L4 big-endian 4-bit

Table 8-292 L4 big-endian 4-bit

Pixel 0				Pixel 1			
7	6	5	4	3	2	1	0
L3	L2	L1	L0	L3	L2	L1	L0
Byte 0							

- L8 8-bit

Table 8-293 L8 8-bit

Pixel 0							
7	6	5	4	3	2	1	0
L7	L6	L5	L4	L3	L2	L1	L0
Byte 0							

- RGBA 8888 32-bit

Table 8-294 RGBA 8888 32-bit

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

- ARGB 8888 32-bit

Table 8-295 ARGB 8888 32-bit

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

- ABGR 8888 32-bit

Table 8-296 ABGR 8888 32-bit

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

- BGRA 8888 32-bit

Table 8-297 BGRA 8888 32-bit

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

- RGBA 5551 16-bit

Table 8-298 RGBA 5551 16-bit

Pixel 0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R4	R3	R2	R1	R0	G4	G3	G2	G1	G1	B4	B3	B2	B1	B0	A0
Byte 1								Byte 0							

- RGBA 565 16-bit

Table 8-299 RGBA 565 16-bit

Pixel 0															
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	G1	B4	B3	B2	B1	B0
Byte 1								Byte 0							

- RGB 332 8-bit

Table 8-300 RGB 332 8-bit

Pixel 0							
7	6	5	4	3	2	1	0
R2	R1	R0	G2	G1	G0	B1	B0
Byte 0							

- TSC4 16-pixel/64-bit

Table 8-301 TSC4 16-pixel/64-bit

Pixel 15 – Pixel 0					
63	62	…	…	1	0
Byte 8 – Byte 0					

- TSC6 16-pixel/96-bit

Table 8-302 TSC6 16-pixel/96-bit

Pixel 15 – Pixel 0					
95	94	…	…	1	0
Byte 12 – Byte 0					

- TSC6A 16-pixel with Alpha/96-bit

Table 8-303 TSC6A 16-pixel with Alpha/96-bit

Pixel 15 – Pixel 0					
95	94	…	…	1	0
Byte 12 – Byte 0					

8.5.4.4.8 Color Output Format

Through SPI, the DC module supports the following output color formats:

- Binary (black & white) 1 bit/pixel
- RGB111 3 bits/pixel
- RGB332 8 bits/pixel
- RGB565 16 bits/pixel
- RGB666 18 bits/pixel
- RGB888 24 bits/pixel

Register INTERFACE_CFG[8:0] bits are used to configure the SPI output color formats, and Table 8-304 shows their values. The [8:6] bits refer to the interface which is set to SPI. The [5:3] bits refer to the data order which can be set from option 0 to option 4. The [2:0] bits refer to the supported color formats which can be set to RGB111, RGB332, RGB444, RGB565, RGB666, and RGB888.

In case of black & white format, users should set the register bits INTERFACE_CFG[2:0] to RGB111 (3 bits/pixel) format and the register bits INTERFACE_CFG[5:3] to option 3. This means that the register bits INTERFACE_CFG[8:0] should be set to 0x0D9 as it can be seen in Table 8-304. The RGB111 format includes four more options that refer to the output data word ordering. The figures in "Section 8.5.4.4.8.1 SPI 3-Wire/4-Wire Output Format" show the SPI output format depending on the configurations of the INTERFACE_CFG[8:0] register bits.

Table 8-304 INTERFACE_CFG[8:0] register bits setup for the SPI interface

INTERFACE_CFG[8:0]	**Interface INTERFACE_CFG[8:6]**	**Option INTERFACE_CFG[5:3]**	**Color Format INTERFACE_CFG[2:0]**
0x0C1	SPI	option 0	RGB111
0x0C9	SPI	option 1	RGB111
0x0D1	SPI	option 2	RGB111
0x0D9	SPI	option 3	RGB111
0x0E1	SPI	option 4	RGB111
0x0C2	SPI	option 0	RGB332
0x0C3	SPI	option 0	RGB444
0x0C5	SPI	option 0	RGB565
0x0C6	SPI	option 0	RGB666

INTERFACE_CFG[8:0]	Interface INTERFACE_CFG[8:6]	Option INTERFACE_CFG[5:3]	Color Format INTERFACE_CFG[2:0]
0x0C7	SPI	option 0	RGB888
0x103	Dual SPI	option 0	RGB444
0x10b	Dual SPI	option 1	RGB444
0x105	Dual SPI	option 0	RGB565
0x106	Dual SPI	option 0	RGB666
0x10e	Dual SPI	option 1	RGB666
0x107	Dual SPI	option 0	RGB888
0x10f	Dual SPI	option 1	RGB888
0x143	Quad SPI	option 0	RGB444
0x145	Quad SPI	option 0	RGB565
0x146	Quad SPI	option 0	RGB666
0x147	Quad SPI	option 0	RGB888

For better screen compatibility, you can do the following:

- Set the delay that begins at the CS activation and ends at the first positive edge of SPI_CLK.
- Set the delay that begins at the last negative edge of SPI_CLK and ends at the CS deactivation.

The settings of the above two parts are based on SPI CLK as the unit, and the range we can choose is as follows:

- 0 = 0 SPI_CLK cycle delay
- 1 = 1 SPI_CLK cycle delay
- 2 = 2 SPI_CLK cycles delay
- 3 = 3 SPI_CLK cycles delay
- 4 = 4 SPI_CLK cycles delay
- 5 = 5 SPI_CLK cycles delay
- 6 = 6 SPI_CLK cycles delay
- 7 = 7 SPI_CLK cycles delay

8.5.4.4.8.1 SPI 3-Wire/4-Wire Output Format

8.5.4.4.8.1.1 Configuration 0x0C1 (RGB111: Option 0)

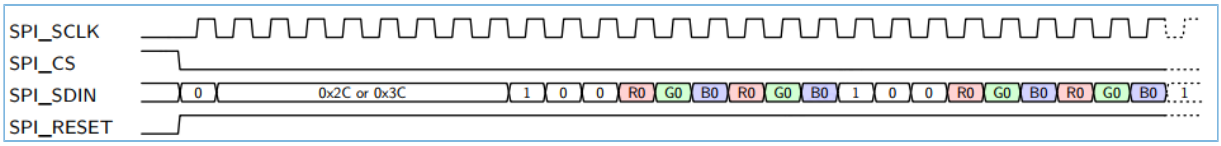

Figure 8-69 SPI-3 RGB111 option 0

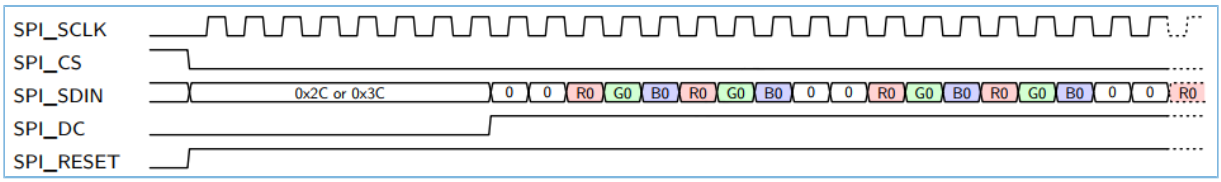

Figure 8-70 SPI-4 RGB111 option 0

8.5.4.4.8.1.2 Configuration 0x0C9 (RGB111: Option 1)

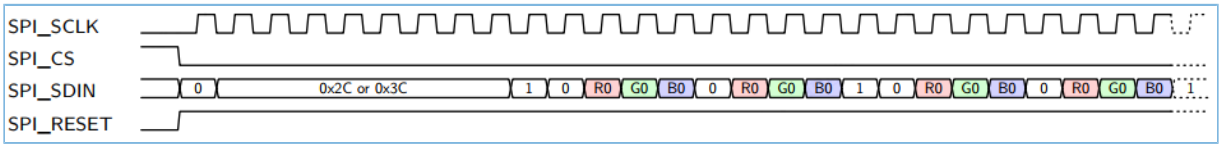

Figure 8-71 SPI-3 RGB111 option 1

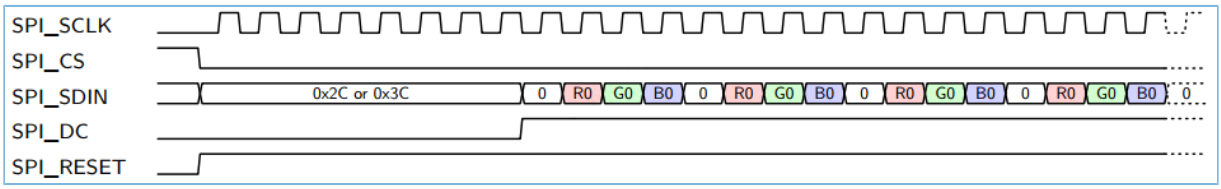

Figure 8-72 SPI-4 RGB111 option 1

8.5.4.4.8.1.3 Configuration 0x0D1 (RGB111: Option 2)

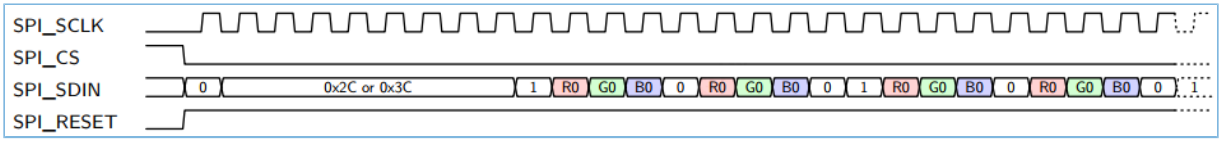

Figure 8-73 SPI-3 RGB111 option 2

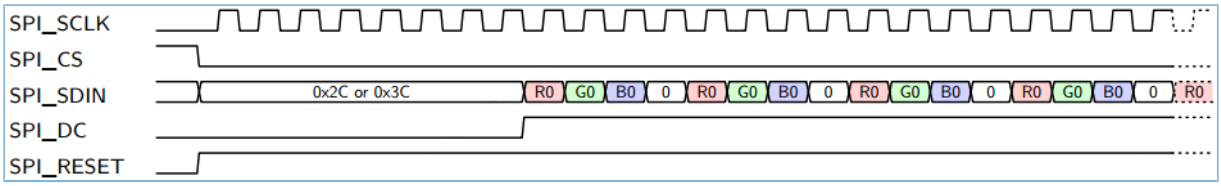

Figure 8-74 SPI-4 RGB111 option 2

8.5.4.4.8.1.4 Configuration 0x0D9 (RGB111: Option 3)

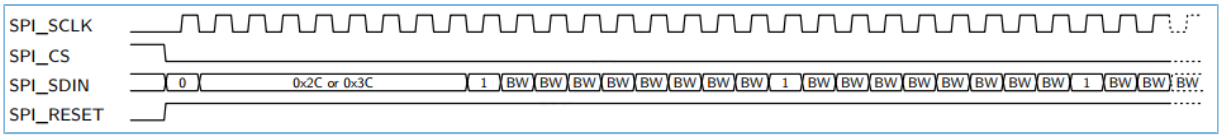

Figure 8-75 SPI-3 RGB111 option 3

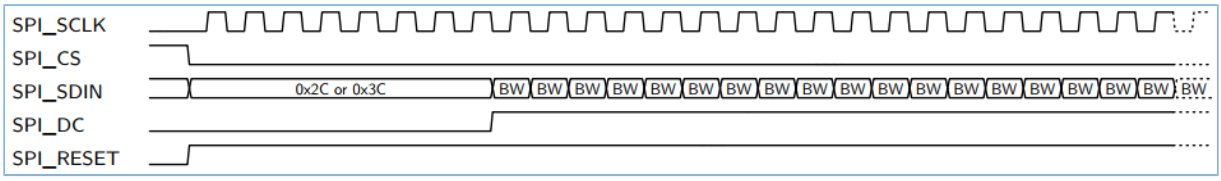

Figure 8-76 SPI-4 RGB111 option 3

8.5.4.4.8.1.5 Configuration 0x0E1 (RGB111: Option 4)

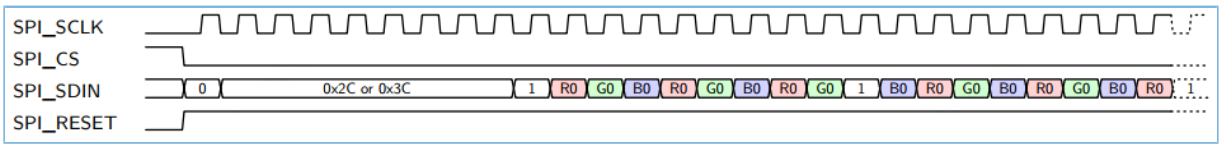

Figure 8-77 SPI-3 RGB111 option 4

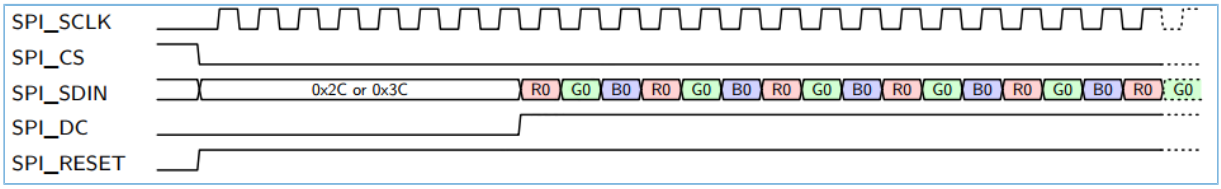

Figure 8-78 SPI-4 RGB111 option 4

8.5.4.4.8.1.6 Configuration 0x0C2 (RGB332-Option 0)

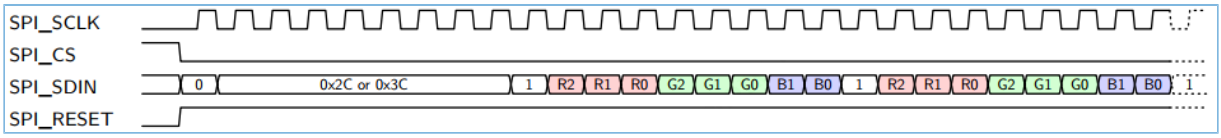

Figure 8-79 SPI-3 RGB332 option 0

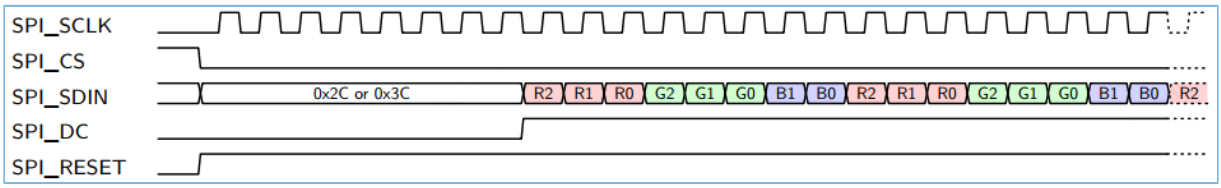

Figure 8-80 SPI-4 RGB332 option 0

8.5.4.4.8.1.7 Configuration 0x0C3 (RGB444-Option 0)

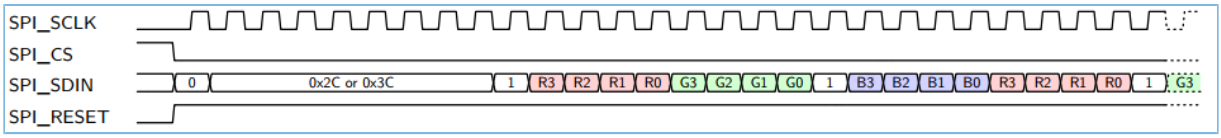

Figure 8-81 SPI-3 RGB444 option 0

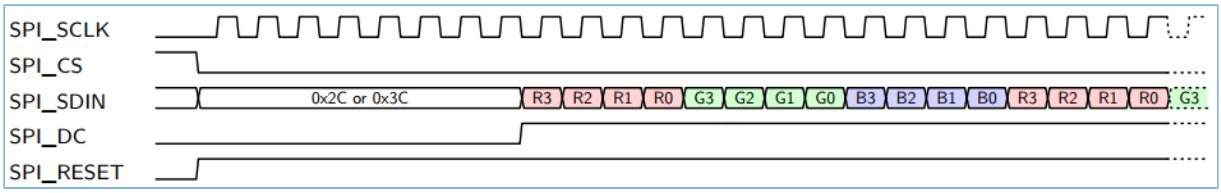

Figure 8-82 SPI-4 RGB444 option 0

8.5.4.4.8.1.8 Configuration 0x0C5 (RGB565-Option 0)

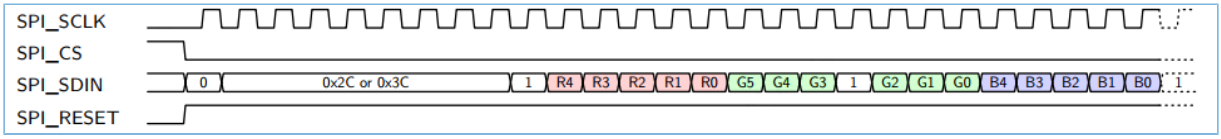

Figure 8-83 SPI-3 RGB565 option 0

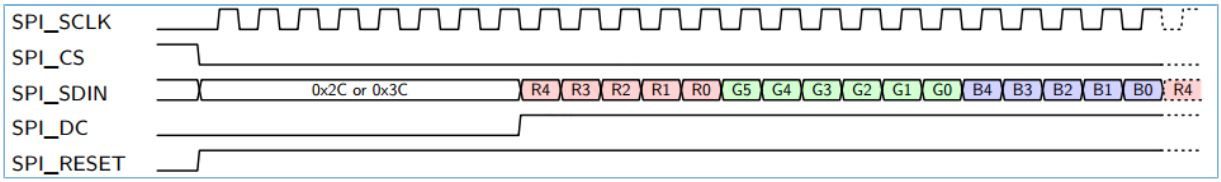

Figure 8-84 SPI-4 RGB565 option 0

8.5.4.4.8.1.9 Configuration 0x0C6 (RGB666-Option 0)

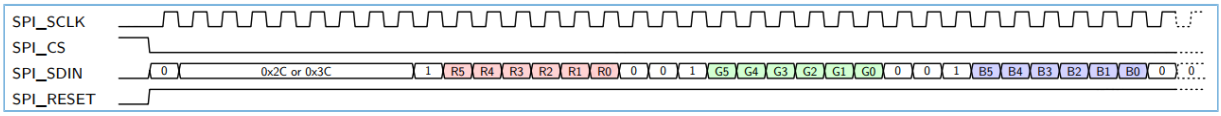

Figure 8-85 SPI-3 RGB666 option 0

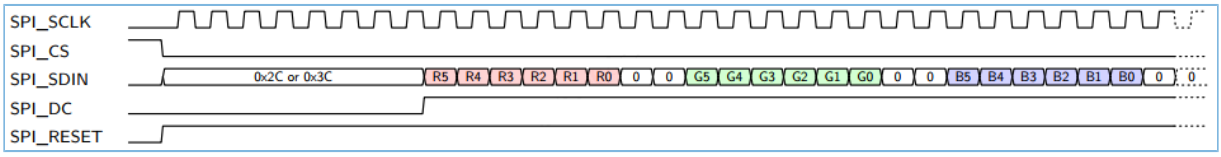

Figure 8-86 SPI-4 RGB666 option 0

8.5.4.4.8.1.10 Configuration 0x0C7 (RGB888-Option 0)

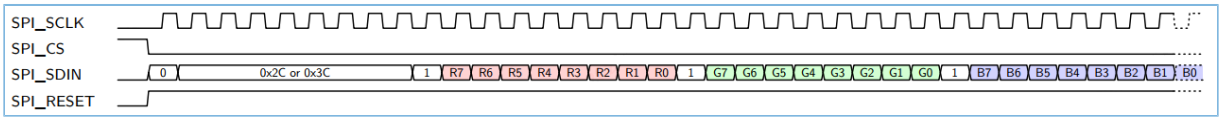

Figure 8-87 SPI-3 RGB888 option 0

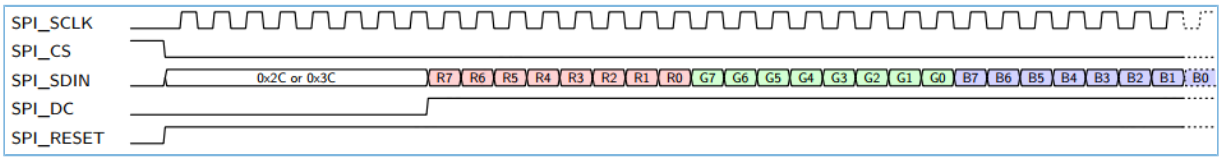

Figure 8-88 SPI-4 RGB888 option 0

8.5.4.4.8.2 DualSPI Output Format

8.5.4.4.8.2.1 Configuration 0x103 (RGB444-Option 0)

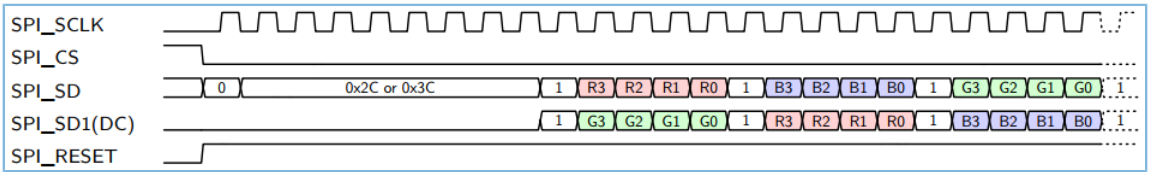

Figure 8-89 DualSPI RGB444 option 0

8.5.4.4.8.2.2 Configuration 0x10B (RGB444-Option 1)

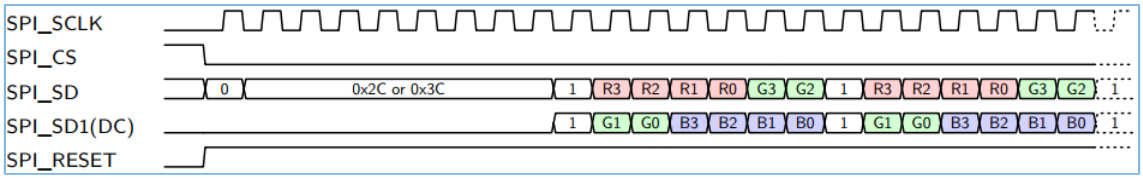

Figure 8-90 DualSPI RGB444 option 1

8.5.4.4.8.2.3 Configuration 0x105 (RGB565-Option 0)

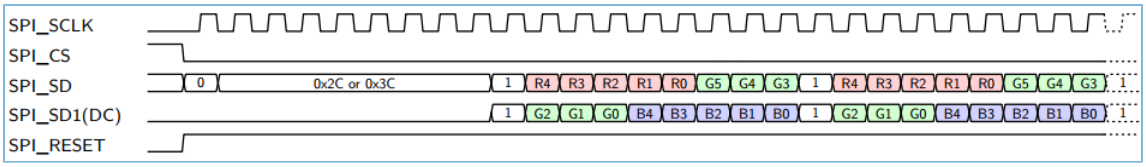

Figure 8-91 DualSPI RGB565 option 0

8.5.4.4.8.2.4 Configuration 0x106 (RGB666-Option 0)

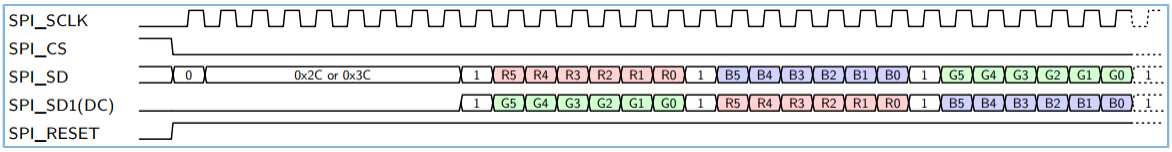

Figure 8-92 DualSPI RGB666 option 0

8.5.4.4.8.2.5 Configuration 0x10E (RGB666-Option 1)

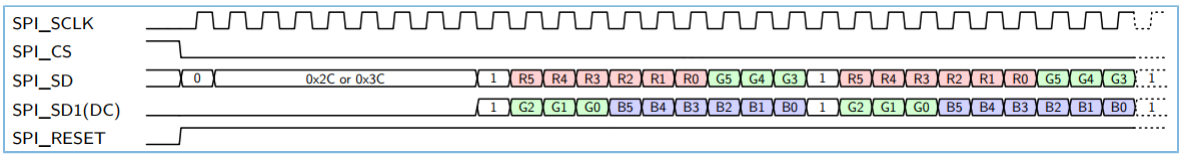

Figure 8-93 DualSPI RGB666 option 1

8.5.4.4.8.2.6 Configuration 0x107 (RGB888-Option 0)

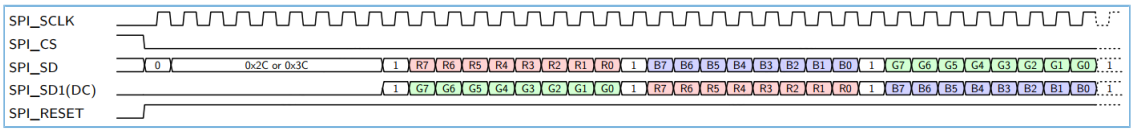

Figure 8-94 DualSPI RGB888 option 0

8.5.4.4.8.2.7 Configuration 0x10F (RGB888-Option 1)

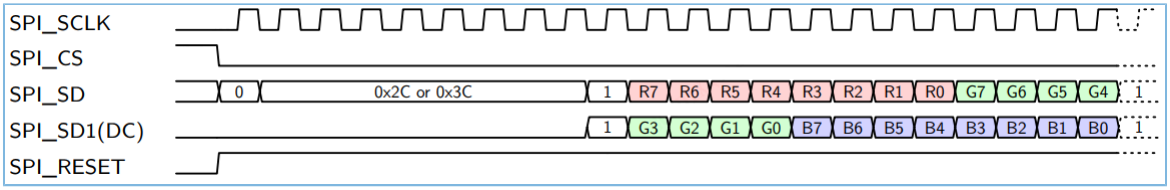

Figure 8-95 DualSPI RGB888 option 1

8.5.4.4.8.3 QuadSPI Serial Interface

The QuadSPI serial interface uses SPI_CS (chip enable), SPI_SCLK (serial clock), as well as SPI_SD, SPI_SD1, SPI_SD2, and SPI_SD3 (serial data outputs). The header and the command type are always transmitted through the SPI_SD line, and the data is routed through the SPI_SD, SPI_SD1, SPI_SD2, and SPI_SD3 lines. The header consists of 8 bits. The command word consists of 24 bits, and the data word consists of n bits according to the chosen format.

8.5.4.4.8.3.1 Configuration 0x143 (RGB444-Option 0)

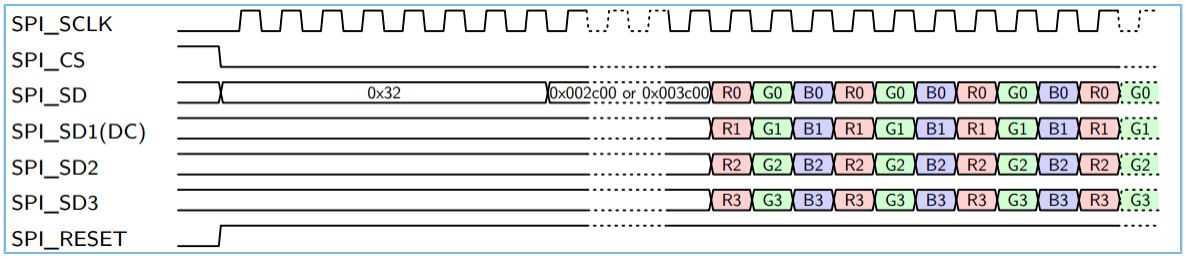

Figure 8-96 QuadSPI RGB444 option 0

8.5.4.4.8.3.2 Configuration 0x145 (RGB565-Option 0)

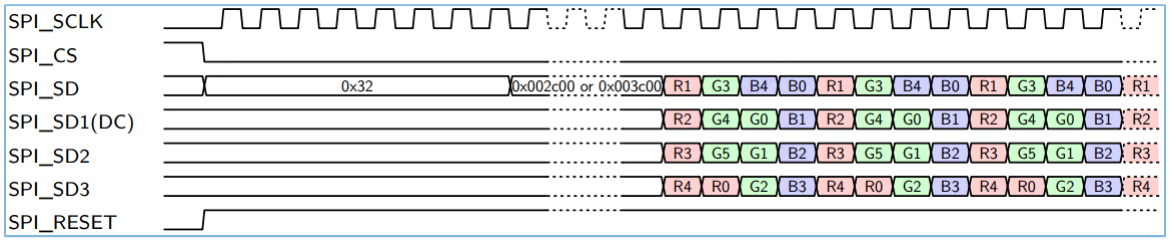

Figure 8-97 QuadSPI RGB565 option 0

8.5.4.4.8.3.3 Configuration 0x146 (RGB666-Option 0)

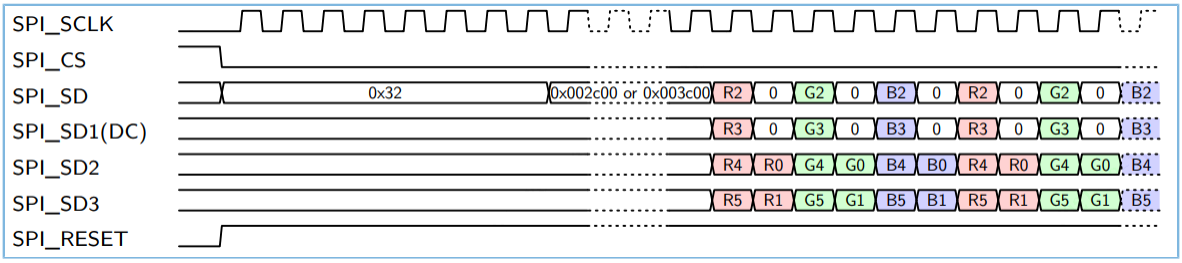

Figure 8-98 QuadSPI RGB666 option 0

8.5.4.4.8.3.4 Configuration 0x147 (RGB888-Option 0)

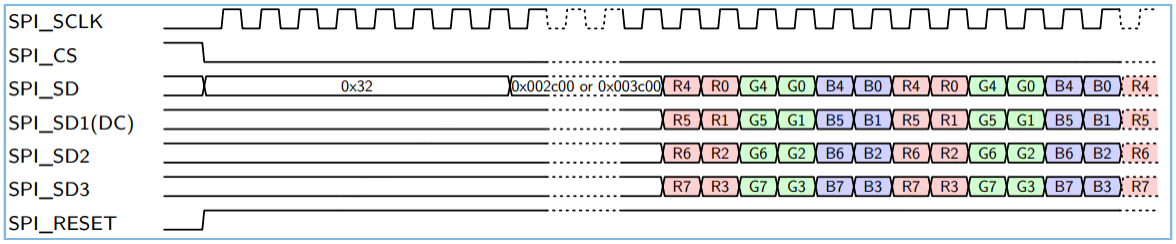

Figure 8-99 QuadSPI RGB888 option 0

8.5.4.5 Registers

8.5.4.5.1 MODE

- **Name:** Mode Register

- **Description:** General control register that activates the DC controller and various parameters, sets the timing signals' polarity, and chooses the output display formats to meet LCD color specifications.
- **Base Address:** 0x4002C000
- **Offset:** 0x000
- **Reset Value:** 0x00000000

Table 8-305 Mode Register

Bits	Field Name	RW	Reset	Description
31	EN	R	0x0	When set to 1, the DC controller is activated.
30:18	RSVD	R		Reserved bits
17	SINGLE_FRAME	RW	0x0	When set to 1, single frame update is enabled.
16:10	RSVD	R		Reserved bits
9	FORMAT	RW	0x0	Selects a format. • 0x0: RGB • 0x1: YUV/YCbCr
8:1	RSVD	R		Reserved bits
0	TEST_MODE	RW	0x0	Test mode • 0x0: Disable • 0x1: Enable

8.5.4.5.2 CLK_DIV

- **Name:** Clock Divider Register
- **Description:** Set proper timing with divisor control bits and specify the number of lines to be prefetched before the start of frame.
- **Base Address:** 0x4002C000
- **Offset:** 0x004
- **Reset Value:** 0x00000401

Table 8-306 Clock Divider Register

Bits	Field Name	RW	Reset	Description
31:27	OPTIONAL_DIV	RW	0x00	Value of optional clock divider • 0x00: No division • 0x01: No division • 0x02: Divided by 2 • 0x0D: Divided by 29

Bits	Field Name	RW	Reset	Description
				• 0x0E: Divided by 30 • 0x0F: Divided by 31
26:6	RSVD	R		Reserved bits
0:5	PRIMARY_DIV	RW	0x00	Value of primary clock divider • 0x00: No division • 0x01: No division • 0x02: Divided by 2 • 0x3D: Divided by 61 • 0x3E: Divided by 62 • 0x3F: Divided by 63

8.5.4.5.3 BG_COLOR

- **Name:** Background Color Register
- **Description:** This register specifies the main background color.
- **Base Address:** 0x4002C000
- **Offset:** 0x008
- **Reset Value:** 0x00000000

Table 8-307 Background Color Register

Bits	Field Name	RW	Reset	Description
31:24	COLOR_BLUE	RW	0x0	Color red is used as background color.
23:16	COLOR_GREEN	RW	0x0	Color green is used as background color.
15:8	COLOR_BLUE	RW	0x0	Color blue is used as background color.
0:7	COLOR_ALPHA	RW	0x0	Color alpha is used as background color.

8.5.4.5.4 RES

- **Name:** X and Y Resolutions Register
- **Description:** This register specifies the main X and Y resolutions.
- **Base Address:** 0x4002C000
- **Offset:** 0x00C
- **Reset Value:** 0x00000000

Table 8-308 X and Y Resolutions Register

Bits	Field Name	RW	Reset	Description
31:16	X_RES	RW	0x0	Value of X resolution in pixels
15:0	Y_RES	RW	0x0	Value of Y resolution in pixels

8.5.4.5.5 PLAY

- **Name:** Play Register
- **Description:** Write any value to this register at the end of the programming sequence to trigger the display controller.
- **Base Address:** 0x4002C000
- **Offset:** 0x010
- **Reset Value:** 0x00000000

Table 8-309 Play Register

Bits	Field Name	RW	Reset	Description
31:0	VALUE	RW	0x0	Trigger the frame generation process.

8.5.4.5.6 INTERFACE_CFG

- **Name:** Interface Configuration Register
- **Description:** The activation of SPI 3-/4-wire interfaces
- **Base Address:** 0x4002C000
- **Offset:** 0x028
- **Reset Value:** 0x00000000

Table 8-310 Interface Configuration Register

Bits	Field Name	RW	Reset	Description
31	INTERFACE_EN	RW	0x0	When set to 1, the SPI interfaces are activated.
30	SW_CS_EN	RW	0x0	When set to 1, the value of the CSX signal of the SPI interfaces can be configured from the INTERFACE_CFG[29] register bit.
29	CS_VAL	RW	0x0	Sets the value of SPI_CS signal: • CS is set to one if INTERFACE_CFG[29] has the value of one. • CS is set to zero if INTERFACE_CFG[29] has the value of zero.

Bits	Field Name	RW	Reset	Description
28:27	RSVD	R		Reserved bits
26	CLEAR_FIFO	RW	0x0	When set to 1, it clears the data FIFO and the command FIFO of the SPI interfaces.
25	RSVD	R		Reserved bit
24	INVERT_PIXEL_ORDER	RW	0x0	Invert pixel order.
23-22	SPI3/4_EN	RW	0x0	Set the SPI 3-wire or 4-wire mode: • 2'b01: SPI 4-wire interface is enabled. • 2'b10: SPI 3-wire interface is enabled.
21	BACK_PRESSURE	RW	0x0	When set to 1, back-pressure for SPI interfaces is enabled.
20	CLK_PHASE	RW	0x0	Sets SPI clock phase.
19	CLK_POLARITY	RW	0x0	Sets SPI clock polarity.
18:12	RSVD	R		Reserved bits
11	DUALSPI_SUB_PIXEL	RW	0x0	When set to 1, DualSPI sub-pixel transaction is enabled.
10:9	MODE	RW	0x0	Set mode of operation- SPI/DualSPI/QuadSPI/QuadSPI DDR: • 0x0: Enables SPI. • 0x1: Enables DualSPI. • 0x2: Enables QuadSPI.
8:6	INTERFAC	RW	0x0	The serial interface(SPI) mode: • 0x3: SPI • 0x4: DualSPI • 0x5: QuadSPI
3:5	DATA_ORDER	RW	0x0	Set the data order of the 8-bit data word: • 0x0: Option 0 • 0x1: Option 1 • 0x2: Option 2 • 0x3: Option 3 • 0x4: Option 4 • 0x5: Reserved • 0x6: Reserved • 0x7: Reserved
2:0	COLOR_FORMAT	RW	0x0	Set the color format for DBI Type-B/SPI interfaces: • 0x1: RGB111 (3 bits/pixel) • 0x2: RGB332 (8 bits/pixel)

Bits	Field Name	RW	Reset	Description
				• 0x3: RGB444 (12 bits/pixel) • 0x5: RGB565 (16 bits/pixel) • 0x6: RGB666 (18 bits/pixel) • 0x7: RGB888 (24 bits/pixel)

8.5.4.5.7 LAYER0_MODE

- **Name:** Layer 0 Mode Register
- **Description:** Activate and set-up layer 0.
- **Base Address:** 0x4002C000
- **Offset:** 0x030
- **Reset Value:** 0x00000000

Table 8-311 Layer 0 Mode Register

Bits	Field Name	RW	Reset	Description
31	LAYER0_EN	RW	0x0	When set to 1, layer n is enabled.
30	FORCE_GLOBAL_ALPHA	RW	0x0	When set to 1, force alpha with global alpha.
29:24	RSVD	R		Reserved bits
23:16	GLOBAL_ALPHA	RW	0x0	Alpha layer global value (range: 0x00-0xFF)
15:12	DST_BLEND	RW	0x0	Destination blending function for layer 0 • 0x00: Blend black • 0x01: Blend white • 0x02: Blend alpha source • 0x03: Blend alpha global • 0x04: Alpha source and alpha global • 0x05:Iinverted source • 0x06: Blend inverted global • 0x07: Inverted source and inverted global • 0x0A: Blend alpha destination • 0x0D: Blend inverted destination
11:8	SRC_BLEND	RW	0x0	Source blending function for layer 0. • 0x00: Blend black • 0x01: Blend white • 0x02: Blend alpha source

Bits	Field Name	RW	Reset	Description
				• 0x03: Blend alpha global • 0x04: Alpha source and alpha global • 0x05:Iinverted source • 0x06: Blend inverted global • 0x07: Inverted source and inverted global • 0x0A: Blend alpha destination • 0x0D: Blend inverted destination
7:6	RSVD	R		Reserved bits
5	REG_PROTECTION	RW	0x0	Enable register protection for DC layer registers. When register protection is enabled, the layer registers take effect in the design only when frame end occurs. Register reading functionality from register file remains the same.
4:0	LAYER0_COLOR_MODE	RW	0x0	Color mode for layer 0. • 0x00: 8-bit color palette look-up table (LUT8) • 0x01: 16-bit RGBA5551 color format • 0x02: 32-bit ABGR8888 color format • 0x04: 8-bit RGB332 color format • 0x05: 16-bit RGB565 color format • 0x06: 32-bit BGRA8888 color format • 0x07: L8 Grayscale/Palette format • 0x08: L1 Black-White/Palette format • 0x09: L4 Grayscale/Palette format • 0x0A: YUYV color format • 0x0B: 24-bit RGB color format • 0x0C: YUY2 color format • 0x0D: 32-bit RGBA8888 color format • 0x0E: 32-bit ARGB8888 color format • 0x10: Video YUV 420 mode • 0x12: TSc4 • 0x13: TSc6 • 0x14: TSc6A

8.5.4.5.8 LAYER0_START

- **Name:** Layer 0 Start Register

- **Description:** X and Y start dimensions of layer 0
- **Base Address:** 0x4002C000
- **Offset:** 0x034
- **Reset Value:** 0x00000000

Table 8-312 Layer 0 Start Register

Bits	Field Name	RW	Reset	Description
31:16	LAYER0_X_POSITION	RW	0x0	Specify the pixel offset of the starting X dimension of layer 0.
15:0	LAYER0_Y_POSITION	RW	0x0	Specify the pixel offset of the starting Y dimension of layer 0.

8.5.4.5.9 LAYER0_SIZE

- **Name:** Layer 0 Size Register
- **Description:** X and Y size of layer 0
- **Base Address:** 0x4002C000
- **Offset:** 0x038
- **Reset Value:** 0x00000000

Table 8-313 Layer 0 Size Register

Bits	Field Name	RW	Reset	Description
31:16	LAYER0_X_SIZE	RW	0x0	Specify the pixel size of the layer 0 in the X dimension.
15:0	LAYER0_Y_SIZE	RW	0x0	Specify the pixel size of the layer 0 in the Y dimension.

8.5.4.5.10 LAYER0_ADDR

- **Name:** Layer 0 Address Register
- **Description:** This register specifies the start address of the framebuffer to be accessed by layer 0.
- **Base Address:** 0x4002C000
- **Offset:** 0x03C
- **Reset Value:** 0x00000000

Table 8-314 Layer 0 Address Register

Bits	Field Name	RW	Reset	Description
31:0	ADDR	RW	0x0	Specify the start address of framebuffer for layer 0.

8.5.4.5.11 LAYER0_STRIDE

- **Name:** Layer 0 Stride Register

- **Description:** This register specifies the stride and the AXI bus burst of layer 0.
- **Base Address:** 0x4002C000
- **Offset:** 0x040
- **Reset Value:** 0x00000000

Table 8-315 Layer 0 Stride Register

Bits	Field Name	RW	Reset	Description
31:26	RSVD	R		Reserved bits
25	DMA_PREFETCH_EN	RW	0x0	DMA prefetch in layer 0 • 0: Disable • 1: Enable
24:17	RSVD	R		Reserved bits
16	INCR16_EN	RW	0x0	When set to 1, INCR16 bursts in layer 0 are allowed.
15:0	Y_POSITION	RW	0x0	Specify the stride, which is the distance from line to line in bytes for layer 0 memory.

8.5.4.5.12 LAYER0_RES

- **Name:** Layer 0 Resolution Register
- **Description:** This register specifies X and Y start dimensions for the resolution of layer 0.
- **Base Address:** 0x4002C000
- **Offset:** 0x044
- **Reset Value:** 0x00000000

Table 8-316 Layer 0 Resolution Register

Bits	Field Name	RW	Reset	Description
31:16	X_RESOLUTION	RW	0x0	Specify the layer 0 pixel resolution in the X dimension.
15:0	Y_RESOLUTION	RW	0x0	Specify the layer 0 pixel resolution in the Y dimension.

8.5.4.5.13 LAYER1_MODE

- **Name:** Layer 1 Mode Register
- **Description:** Activate and set-up layer 1.
- **Base Address:** 0x4002C000
- **Offset:** 0x050
- **Reset Value:** 0x00000000

Table 8-317 Layer 1 Mode Register

Bits	Field Name	RW	Reset	Description
31	LAYER1_EN	RW	0x0	When set to 1, layer n is enabled.
30	FORCE_GLOBAL_ALPHA	RW	0x0	When set to 1, force alpha with global alpha.
29:24	RSVD	R		Reserved bits
23:16	GLOBAL_ALPHA	RW	0x0	Alpha layer global value (range: 0x00-0xFF)
15:12	DST_BLEND	RW	0x0	Destination blending function for layer 1. • 0x00: Blend black • 0x01: Blend white • 0x02: Blend alpha source • 0x03: Blend alpha global • 0x04: Alpha source and alpha global • 0x05:Iinverted source • 0x06: Blend inverted global • 0x07: Inverted source and inverted global • 0x0A: Blend alpha destination • 0x0D: Blend inverted destination
11:8	SRC_BLEND	RW	0x0	Source blending function for layer 1. • 0x00: Blend black • 0x01: Blend white • 0x02: Blend alpha source • 0x03: Blend alpha global • 0x04: Alpha source and alpha global • 0x05:Iinverted source • 0x06: Blend inverted global • 0x07: Inverted source and inverted global • 0x0A: Blend alpha destination • 0x0D: Blend inverted destination
7:6	RSVD	R		Reserved bits
5	REG_PROTECTION	RW	0x0	Enable register protection for DC layer registers. When register protection is enabled, the layer registers take effect in the design only when frame end occurs. Register reading functionality from register file remains the same.
4:0	LAYER1_COLOR_MODE	RW	0x0	Color mode for layer 1.

Bits	Field Name	RW	Reset	Description
				• 0x00: 8-bit color palette look-up table (LUT8) • 0x01: 16-bit RGBA5551 color format • 0x02: 32-bit ABGR8888 color format • 0x04: 8-bit RGB332 color format • 0x05: 16-bit RGB565 color format • 0x06: 32-bit BGRA8888 color format • 0x07: L8 Grayscale/Palette format • 0x08: L1 Black-White/Palette format • 0x09: L4 Grayscale/Palette format • 0x0A: YUYV color format • 0x0B: 24-bit RGB color format • 0x0C: YUY2 color format • 0x0D: 32-bit RGBA8888 color format • 0x0E: 32-bit ARGB8888 color format • 0x10: Video YUV 420 mode • 0x12: TSc4 • 0x13: TSc6 • 0x14: TSc6A

8.5.4.5.14 LAYER1_START

- **Name:** Layer 1 Start Register
- **Description:** X and Y start dimensions of layer 1
- **Base Address:** 0x4002C000
- **Offset:** 0x054
- **Reset Value:** 0x00000000

Table 8-318 Layer 1 Start Register

Bits	Field Name	RW	Reset	Description
31:16	LAYER1_X_POSITION	RW	0x0	Specify the pixel offset of the starting X dimension of layer 1.
15:0	LAYER1_Y_POSITION	RW	0x0	Specify the pixel offset of the starting Y dimension of layer 1.

8.5.4.5.15 LAYER1_SIZE

- **Name:** Layer 1 Size Register

- **Description:** X and Y size of layer 1
- **Base Address:** 0x4002C000
- **Offset:** 0x058
- **Reset Value:** 0x00000000

Table 8-319 Layer 1 Size Register

Bits	Field Name	RW	Reset	Description
31:16	LAYER1_X_SIZE	RW	0x0	Specify the pixel size of the layer 1 in the X dimension.
15:0	LAYER1_Y_SIZE	RW	0x0	Specify the pixel size of the layer 1 in the Y dimension.

8.5.4.5.16 LAYER1_ADDR

- **Name:** Layer 1 Address Register
- **Description:** This register specifies the start address of the framebuffer to be accessed by layer 1.
- **Base Address:** 0x4002C000
- **Offset:** 0x05C
- **Reset Value:** 0x00000000

Table 8-320 Layer 1 Address Register

Bits	Field Name	RW	Reset	Description
31:0	ADDR	RW	0x0	Specify the start address of framebuffer for layer 1.

8.5.4.5.17 LAYER1_STRIDE

- **Name:** Layer 1 Stride Register
- **Description:** This register specifies the stride and the AXI bus burst of layer 1.
- **Base Address:** 0x4002C000
- **Offset:** 0x060
- **Reset Value:** 0x00000000

Table 8-321 Layer 1 Stride Register

Bits	Field Name	RW	Reset	Description
31:26	RSVD	R		Reserved bits
25	DMA_PREFETCH_EN	RW	0x0	DMA prefetch in layer 1 • 0: Disable • 1: Enable
24:17	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
16	INCR16_EN	RW	0x0	When set to 1, INCR16 bursts in layer 1 are allowed.
15:0	Y_POSITION	RW	0x0	Specify the stride, which is the distance from line to line in bytes for layer 1 memory.

8.5.4.5.18 LAYER1_RES

- **Name:** Layer 1 Resolution Register
- **Description:** This register specifies X and Y start dimensions for the resolution of layer 1.
- **Base Address:** 0x4002C000
- **Offset:** 0x064
- **Reset Value:** 0x00000000

Table 8-322 Layer 1 Resolution Register

Bits	Field Name	RW	Reset	Description
31:16	X_RESOLUTION	RW	0x0	Specify the layer 1 pixel resolution in the X dimension.
15:0	Y_RESOLUTION	RW	0x0	Specify the layer 1 pixel resolution in the Y dimension.

8.5.4.5.19 INTERFACE_CMD

- **Name:** Interface Command Register
- **Description:** Command register to read/write commands from/to SPI interfaces
- **Base Address:** 0x4002C000
- **Offset:** 0x0E8
- **Reset Value:** 0x00000000

Table 8-323 Interface Command Register

Bits	Field Name	RW	Reset	Description
31	BASE_ADDR_FLAG	RW	0x0	Being set to 0 indicates that the command data is the base address for partial update. Applied on SPI type.
30	SEND_DATA	RW	0x0	Send direct data of type command to the SPI interfaces.
29:28	CMD_WIDTH	RW	0x0	Determine the command width. Applicable only to QuadSPI. • 0x00: 1 byte • 0x01: 2 bytes • 0x02: 3 bytes • 0x03: Reserved

Bits	Field Name	RW	Reset	Description
27	QSPI_SERIAL_EN	RW	0x0	Being set to 1 means switching to serial transmission of the command. Applicable only to QuadSPI.
26	READ_EN	RW	0x0	When set to 1, read mode is enabled.
25	RSVD	R		Reserved bit
24	CMD_LINE_EN	RW	0x0	When set to 1, store internally a command type which is transmitted at the beginning of each scanline.
23:16	RSVD	R		Reserved bits
15:0	DATA	RW	0x0	Data to be sent to the SPI interfaces

8.5.4.5.20 INTERFACE_RDAT

- **Name:** Interface Read Data Register
- **Description:** Data read by DBI Type-B/SPI interfaces is stored in the register.
- **Base Address:** 0x4002C000
- **Offset:** 0x0EC
- **Reset Value:** 0x00000000

Table 8-324 Interface Read Data Register

Bits	Field Name	RW	Reset	Description
31:30	READ_CYCLES	RW	0x0	On write: Specify the number of read cycles. • 0x0: 8 cycles • 0x1: 16 cycles • 0x2: 24 cycles • 0x3: Used along with FMTCTRL[21:16] register bits On read: Read data from DBI Type-B/SPI interfaces (high bits).
29:0	READ_DATA	R	0x0	Read data from DBI Type-B/SPI interfaces (low bits).

8.5.4.5.21 CFG

- **Name:** Configuration Register
- **Description:** Information register of the layers activation and setup. The reset value of this register changes according to the selected configuration.
- **Base Address:** 0x4002C000
- **Offset:** 0x0F0

- **Reset Value:** 0x01003130

Table 8-325 Configuration Register

Bits	Field Name	RW	Reset	Description
31	LAYER3_YUV_MEM_EN	R	0x0	Indicates that layer 3 has YUV Memory.
30	LAYER2_YUV_MEM_EN	R	0x0	Indicates that layer 2 has YUV Memory.
29	LAYER1_YUV_MEM_EN	R	0x0	Indicates that layer 1 has YUV Memory.
28	LAYER0_YUV_MEM_EN	R	0x0	Indicates that layer 0 has YUV Memory.
27:25	RSVD	R		Reserved bits
24	SPI_EN	R	0x1	Indicates that SPI interface is enabled.
23	LAYER3_GAMMA_EN	R	0x0	Indicates that layer 3 has Gamma LUT.
22	RSVD	R		Reserved bit
21	LAYER3_BLENDER_EN	R	0x0	Indicates that layer 3 has blender.
20	LAYER3_EN	R	0x0	Indicates that layer 3 is enabled.
19	LAYER2_GAMMA_EN	R	0x0	Indicates that layer 2 has Gamma LUT.
18	RSVD	R		Reserved bit
17	LAYER2_BLENDER_EN	R	0x0	Indicates that layer 2 has blender.
16	LAYER2_EN	R	0x0	Indicates that layer 2 is enabled.
15	LAYER1_GAMMA_EN	R	0x0	Indicates that layer 1 has Gamma LUT.
14	RSVD	R		Reserved bit
13	LAYER1_BLENDER_EN	R	0x1	Indicates that layer 1 has blender.
12	LAYER1_EN	R	0x1	Indicates that layer 1 is enabled.
11	LAYER0_GAMMA_EN	R	0x0	Indicates that layer 0 has Gamma LUT.
10	RSVD	R		Reserved bit
9	LAYER0_BLENDER_EN	R	0x0	Indicates that layer 0 has blender.
8	LAYER0_EN	R	0x1	Indicates that layer 0 is enabled.
7	RGB_TO_YUV_EN	R	0x0	Indicates that RGB to YUV/YCbCr converter is enabled.
6	DBI_TYPEB_EN	R	0x0	Indicates that DBI Type-B interface is enabled.
5	YUV_CONVERTER_EN	R	0x1	Indicates that high quality YUV/YCbCr converter is enabled.
4	FORMAT_EN	R	0x1	Indicates that formatting is enabled.
3	DITHER_EN	R	0x0	Indicates that dithering is enabled.
2	RSVD	R		Reserved bit
1	CURSOR_EN	R	0x0	Indicates that cursor is enabled.
0	GAMMA_PALETTE_EN	R	0x0	Indicates that Global Gamma/Palette is enabled.

8.5.4.5.22 ID

- **Name:** Identification Register
- **Description:** This register specifies the identification of the DC.
- **Base Address:** 0x4002C000
- **Offset:** 0x0F4
- **Reset Value:** 0x87452365

Table 8-326 Identification Register

Bits	Field Name	RW	Reset	Description
31:0	ID	R	0x87452365	Constant ID value

8.5.4.5.23 INT

- **Name:** Interrupt Register
- **Description:** This register specifies the interrupt of DC.
- **Base Address:** 0x4002C000
- **Offset:** 0x0F8
- **Reset Value:** 0x00000001

Table 8-327 Interrupt Register

Bits	Field Name	RW	Reset	Description
31	IRQ_TRIGGER	R	0x0	Interrupt request trigger control • 0: Level triggering is enabled • 1: Edge triggering is enabled
30	IRQ_UNDERWAY	R	0x0	Indicates IRQ is underway.
29:7	RSVD	R		Reserved bits
6	CMD_INT_EN	RW	0x0	When set to 1, signal an interrupt when SPI/DBI interfaces have finished the transmission of the commands.
5	FRAME_CPLT_INT_EN	RW	0x0	When set to 1, signal an interrupt when SPI/DBI interfaces have finished the transmission of the frame.
4	FRMAE_END_INT_EN	RW	0x0	When set to 1, signal an interrupt on frame end.
3	TE_INT_EN	RW	0x0	When set to 1, TE interrupt is enabled.
2	RSVD	R		Reserved bit
1	HSYNC_INT_EN	RW	0x0	When set to 1, HSYNC interrupt is enabled.
0	VSYNC_INT_EN	RW	0x1	When set to 1, VSYNC interrupt is enabled.

8.5.4.5.24 STAT

- **Name:** Status Register
- **Description:** This register specifies the status of DC.
- **Base Address:** 0x4002C000
- **Offset:** 0x0FC
- **Reset Value:** 0x00000000

Table 8-328 Status Register

Bits	Field Name	RW	Reset	Description
31:19	RSVD	R		Reserved bits
18	GPI_STACK	R	0x0	Indicates GPI gets stack during packet transaction.
17	CRC_READY	R	0x0	Indicates CRC register read is ready.
16	SPI_RW	R	0x0	Indicates read/write operation on SPI.
15	FIFO_ACK	R	0x0	Indicates the DBI/SPI command FIFO ACK.
14	CS_STATUS	R	0x0	Indicates DBI/SPI CS status.
13	FRAME_END	R	0x0	Indicates frame ends.
12	SPI_PENDING	R	0x0	Indicates pending output transaction in DBI Type-B/SPI interfaces.
11	CMD_PENDING	R	0x0	Indicates pending commands in DBI Type-B/SPI interfaces.
10	DATA_PENDING	R	0x0	Indicates pending RGB data in DBI Type-B/SPI interfaces.
9	RSVD	R		Reserved bit
8	TEARING_EFFECT	R	0x0	Indicates DBI Type-B/SPI tearing effect.
7	STICKY_UNDERFLOW	R	0x0	Indicates sticky underflow. This bit is cleared when interrupt register is written.
6	CUR_UNDERFLOW	R	0x0	Indicates current underflow.
5	LAST_LOW	R	0x0	Indicates that the last row is currently displayed.
4	CSYNC_STAT	R	0x0	Indicates the CSYNC signal status (0 or 1) at the current time of reading.
3	VSYNC_STAT	R	0x0	Indicates the VSYNC signal status and the tearing effect signal status (0 or 1) at the current time of reading.
2	HSYNC_STAT	R	0x0	Indicates the HSYNC signal status (0 or 1) at the current time of reading.
1	DE_STAT	R	0x0	Indicates the DE signal status (0 or 1) at the current time of reading.
0	DC_STAT	R	0x0	Indicates that the controller is not in active vertical blanking.

8.5.4.5.25 COLOR_MODE

- **Name:** Color Mode Register
- **Description:** This register specifies the color mode of DC.
- **Base Address:** 0x4002C000
- **Offset:** 0x100
- **Reset Value:** 0x0067FF59

Table 8-329 Color Mode Register

Bits	Field Name	RW	Reset	Description
31:29	RSVD	R		Reserved bits
28:23	RSVD	R		Reserved bits
22	ARGB4444	R	0x1	Setting to zero indicates that the ARGB4444 16-bit color format is disabled.
21	RGBA4444	R	0x1	Setting to zero indicates that the RGBA4444 16-bit color format is disabled.
20:19	RSVD	R		Reserved bits
18	TSC6	R	0x1	High value indicates that TSC6 is disabled.
17	TSC4	R	0x1	High value indicates that TSC4 is disabled.
16	LUT8	R	0x1	Setting to zero indicates that the LUT8 color format is disabled.
15	RGBA5551	R	0x1	Setting to zero indicates that the RGBA5551 16-bit color format is disabled.
14	ABGR8888	R	0x1	Setting to zero indicates that the ABGR8888 32-bit color format is disabled.
13	RGB332	R	0x1	Setting to zero indicates that the RGB332 8-bit color format is disabled.
12	RGB565	R	0x1	Setting to zero indicates that the RGB565 16-bit color format is disabled.
11	ARGB8888	R	0x1	Setting to zero indicates that the ARGB8888 32-bit color format is disabled.
10	L8	R	0x1	Setting to zero indicates that the L8 color format is disabled.
9	L1	R	0x1	Setting to zero indicates that the L1 color format is disabled.
8	L4	R	0x1	Setting to zero indicates that the L4 color format is disabled.
7	YUYV	R	0x0	Setting to zero indicates that the YUYV color format is disabled.

Bits	Field Name	RW	Reset	Description
6	RGB888	R	0x1	Setting to zero indicates that the RGB888 24-bit color format is disabled.
5	YUY2	R	0x0	Setting to zero indicates that the YUY2 color format is disabled.
4	RGBA8888	R	0x1	Setting to zero indicates that the RGBA8888 32-bit color format is disabled.
3	BGRA8888	R	0x1	Setting to zero indicates that the BGRA8888 32-bit color format is disabled.
2	V_YUV420	R	0x0	Setting to zero indicates that the V_YUV420 color format is disabled.
1	TLYUV420	R	0x0	Setting to zero indicates that the TLYUV420 color format is disabled.
0	TSC4_TSC6	R	0x1	Setting to zero indicates that the TSC4/TSC6 proprietary color format is disabled.

8.5.4.5.26 CRC

- **Name:** CRC Check Register
- **Description:** This register specifies the CRC value.
- **Base Address:** 0x4002C000
- **Offset:** 0x184
- **Reset Value:** 0x00000000

Table 8-330 CRC Check Register

Bits	Field Name	RW	Reset	Description
31:0	CRC	R	0x0	CRC value if CRC error exists

8.5.4.5.27 CLK_CTRL

- **Name:** Clock Control Register
- **Description:** Controls the clock gaters and the routing of format and pixel clock.
- **Base Address:** 0x4002C000
- **Offset:** 0x1A8
- **Reset Value:** 0x00000000

Table 8-331 Clock Control Register

Bits	Field Name	RW	Reset	Description
31:28	RSVD	R		Reserved bits
27	LAYER1_BUS_CLK_BYPASS	RW	0x0	Layer 1 bus clock clock-gater bypass
26	LAYER1_PIXEL_CLK_BYPASS	RW	0x0	Layer 1 pixel clock clock-gater bypass
25	LAYER0_BUS_CLK_BYPASS	RW	0x0	Layer 0 bus clock clock-gater bypass
24	LAYER0_PIXEL_CLK_BYPASS	RW	0x0	Layer 0 pixel clock clock-gater bypass
23	REGFILE_BYPASS	RW	0x0	RegFile clock-gaters bypass
22	BYPASS	RW	0x0	Clock-gaters bypass
21:3	RSVD	R		Reserved bits
2	CLK_SWAP	RW	0x0	Pixel generation and format clock swap
1	CLK_POLARITY	RW	0x0	Invert (output) clock polarity.
0	CLK_DIV_EN	RW	0x0	Enable clock divider.

8.5.4.5.28 FORMAT_CTRL

- **Name:** CS Format Control Register
- **Description:** Control CS format.
- **Base Address:** 0x4002C000
- **Offset:** 0x1AC
- **Reset Value:** 0x00000000

Table 8-332 CS Format Control Register

Bits	Field Name	RW	Reset	Description
31:16	RSVD	R		Reserved bits
15:13	CS_END_DELAY	RW	0x0	Set the delay that begins at the last negative edge of SPI_CLK and ends at the CS deactivation. Values of bits [15:13] while reading from a panel in SDR mode and chip phase is 1: • 0: Zero SPI_CLK cycle delay • 1: One SPI_CLK cycle delay • 2: Two SPI_CLK cycles delay • 3: Three SPI_CLK cycles delay • 4: Four SPI_CLK cycles delay • 5: Five SPI_CLK cycles delay

Bits	Field Name	RW	Reset	Description
				• 6: Six SPI_CLK cycles delay • 7: Seven SPI_CLK cycles delay
12:10	CS_BEGIN_DELAY	RW	0x0	Set the delay that begins at the CS activation and ends at the first positive edge of SPI_CLK. Values of bits [12:10] in SDR (Single Data Rate) and DDR (Dual Data Rate) modes: • 0: Zero SPI_CLK cycle delay • 1: One SPI_CLK cycle delay • 2: Two SPI_CLK cycles delay • 3: Three SPI_CLK cycles delay • 4: Four SPI_CLK cycles delay • 5: Five SPI_CLK cycles delay • 6: Six SPI_CLK cycles delay • 7: Seven SPI_CLK cycles delay The above equations apply to both write and read modes regardless the values of chip phase and polarity.
9:0	RSVD	R		Reserved bits

8.5.4.6 Electrical Specifications

The timing of this the DC module meets the timing standard of MIPI type C, as follows:

Table 8-333 DC electrical specifications

Parameter	Description	Min.	Typ.	Max.	Unit
f_{sck}	Clock frequency			48	M
t_{css}	CSX Chip Select setup time(Write)	4T			ns
t_{csh}	CSX Chip Select setup time(Read)	4T			ns
t_{as}	D/CX Address setup time	T			ns
t_{ah}	D/CX Address hold time	T			ns
t_{wc}	SCL Write cycle	10T			ns
t_{wrh}	SCL H duration(Write)	4T			ns
t_{wrl}	SCL L duration(Write)	4T			ns
t_{rc}	SCL Read cycle	15T			ns
t_{rdh}	SCL H duration(Read)	6T			ns
t_{rdl}	SCL L duration(Read)	6T			ns

Parameter	Description	Min.	Typ.	Max.	Unit
t_{ds}	Write data setup time	15			ns
t_{dh}	Write data hold time	25			ns
t_{acc}	Access time	10			ns
t_{od}	Output disable time	T			ns

Note:

Temperature = -30°C to 70°C; VDDI range: dependent on Logic High level input voltage classification; GND = 0 V; T = 10 (±0.5) ns

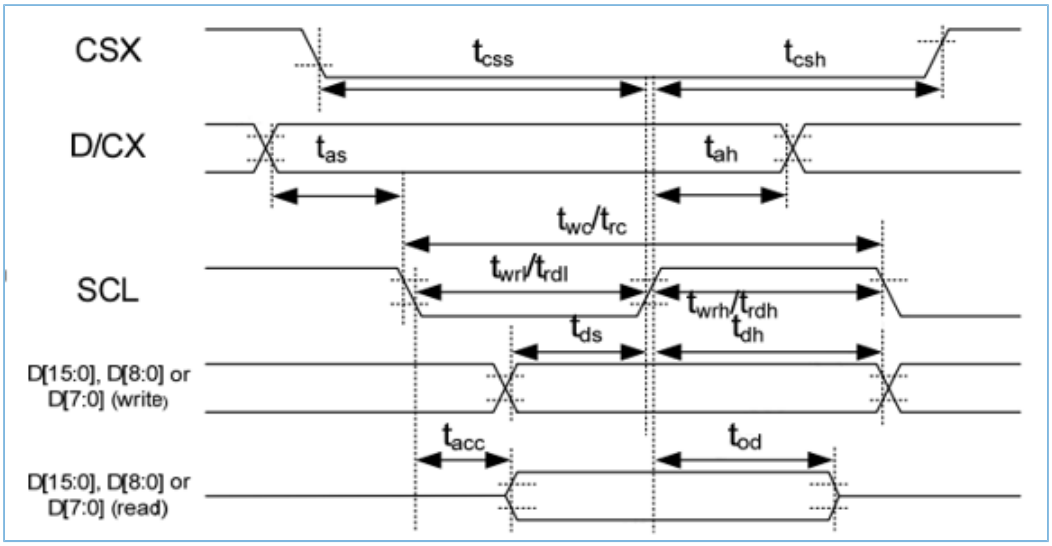

Figure 8-100 DC timing diagram

8.5.5 GPU

8.5.5.1 Introduction

Graphics Processing Unit (GPU) is a kind of microprocessor that specializes in image and graphic related operations. GR5526 integrates a low-power GPU which can bring high-quality graphics to the user interface with a very small silicon/power budget. It supports entry-level IoT platforms, wearables, and embedded devices with low-cost and ultra-low power consumption requirements, and provides a smooth graphical experience suitable for various applications. Developers can create compelling graphical user interfaces (GUIs) and software applications with extremely long battery life for power-memory-area constrained IoT devices at significantly lower cost.

8.5.5.2 Main Features

- Hardware components:
  - Programmable shader engine with a very long instruction word (VLIW) architecture
  - Command-list-based DMAs to minimize CPU overhead

  - Primitive rasterizer
  - Texture mapping unit
  - Blending unit
- Drawing primitives:
  - Pixel/Line drawing
  - Filled rectangles
  - Triangles (Gouraud shaded)
  - Quadrilaterals
- Color formats:
  - 32-bit RGBA8888/BGRA8888/ABGR8888
  - 24-bit RGB
  - 16-bit RGBA5551/RGB565
  - 8-bit A8/L8/RGB332
  - TSC™
- Compression & Decompression feature:
  - Image transformation
  - Texture mapping
  - Point sampling
  - Bilinear filtering
  - Blit support
  - Rotation any angle
  - Mirroring
  - Stretch (independently on X and Y axes)
  - Source and/or destination color keying
  - Format conversions on the fly
  - 2.5D perspective correct projections
- Text rendering support:
  - Bitmap antialiased A1/A2/A4/A8
  - Font Kerning
  - Unicode (UTF8)

- Blending support:
  - Fully programmable Alpha blending modes (source and destination)
  - Source/Destination color keying
- Antialiasing:
  - 8 x MSAA
  - Quadrilaterals per edge antialiasing
  - Triangles per edge antialiasing
  - Antialiased thick lines
  - Antialiased circles

8.5.5.3 Functional Description

8.5.5.3.1 Block Diagram

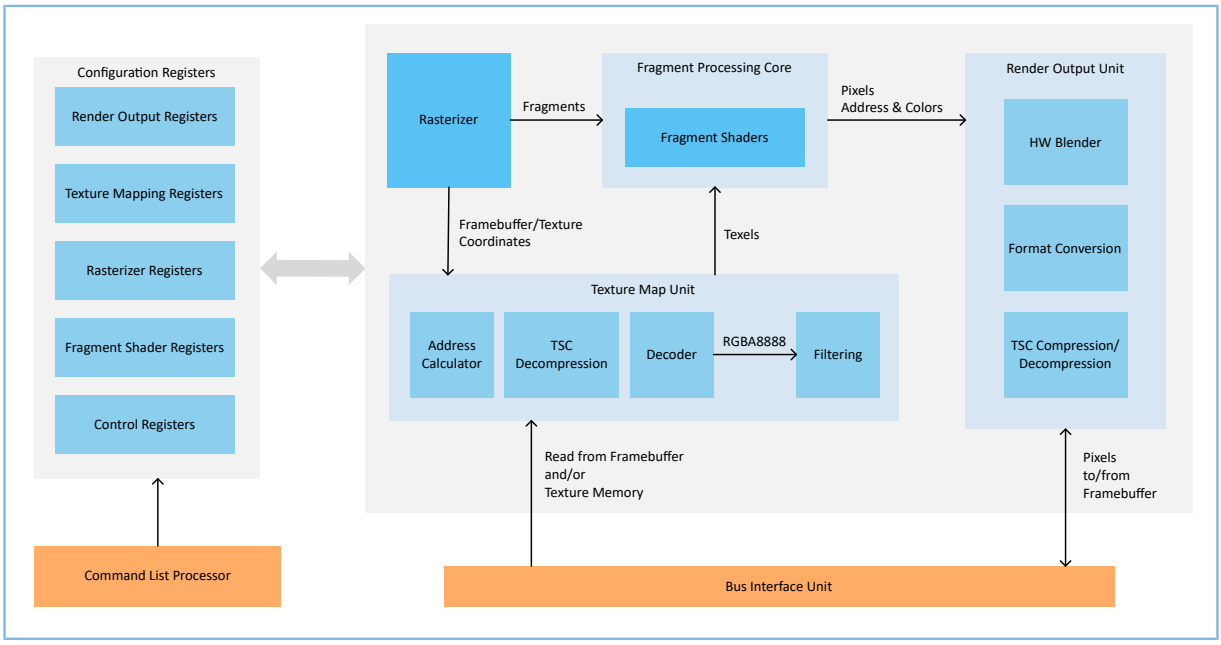

Figure 8-101 GPU block diagram

The fixed-point data path and instruction set architecture (ISA) of the GPU are tailored specifically for GUIs acceleration and small display applications, resulting in significant improvements in power consumption and silicon area. The GPU combines hardware-level support for multithreading, VLIW, and low-level vector processing in the most energy-efficient way. The GPU block diagram is shown above.

8.5.5.3.2 I/O Interfaces

The GPU supports connection via AHB that helps it communicate with the MCU, configuration registers, and system memory. The GPU uses two 32-bit AHB master ports that enable access to the main memory and that support data

fetch from the G PU (such as textures and frame data). The AHB version of the system fetches both the command lists and rendering data over the same read bus.

8.5.5.3.3 Configuration Register File

The GPU is programmed through a set of registers which are known as configuration register file (CRF), and each sub-module of the GPU is programmed through a subset of the CRF. CRF can be memory-mapped to the CPU address space, making it directly accessible. Writing CRF directly is considered inefficient because it consumes a lot of CPU resources and associates CPU execution with the GPU. Therefore, the GPU supports indirect CRF access through the Command List Processor (CLP).

8.5.5.3.4 Command List Processor

In order to decouple CPU from GPU execution and achieve both better performance and lower power consumption, the GPU incorporates an advanced Command List Processor (CLP), capable of reading entire lists of commands from the main memory and relaying them to the CRF.

The CPU pre-assembles Command Lists (CLs) prior to submitting them to the CLP for execution, while a single CL can be submitted multiple times. This approach alleviates the CPU from recalculating drawing operations for repetitive tasks, resulting in more efficient resource utilization.

The steps for writing commands to the configuration registers through the CLP are the following:

1. The CPU assembles a CL, through the GFX library.
2. The CPU submits the CL for execution. The CLP is informed of a pending CL.
3. The CLP reads the CL from the system memory.
4. The CLP relays the commands to the Configuration Register File.

8.5.5.3.5 Rasterizer

The GPU can draw a multitude of geometrical shapes called Geometric Primitives, such as lines, rectangles, triangles, and quadrilaterals. The rasterizer unit reads the coordinates of the primitives' vertices and feeds the rest of the graphics pipeline with the fragments contained in the geometry. A fragment contains information concerning a single pixel. This information includes raster position (coordinates), texture coordinates, interpolated color, and alpha values.

The rasterizer can draw:

- Line drawing (at any direction)
- Filled rectangles
- Quadrilaterals
- Triangles

In addition, the rasterizer handles clipping by either dropping fragments that are outside the effective drawing area and back-face culling or dropping entire primitives that are considered to be non-visible, like the rear looking faces of a cube. If a pixel resides inside or outside of the geometry primitive, the pixel is determined by the value of E which is positive inside the geometry primitive and negative outside of it.

The pixel's edge function value is calculated for each line of the geometry primitive (for example, 3 times for a triangle) using the following equation:

$$E = A \cdot x + B \cdot y + C \tag{24}$$

Although the equation requires two multiplications and two additions since the variation is always one pixel on the X or Y axis, this is reduced to a single accumulator.

The color variances are also calculated in a similar way, where each of the RGBA components is linearly interpolated across the geometry of the primitive. If H/W Blender is enabled, each edge can be independently programmed to either have antialiasing or not. The rasterizer also determines the coverage value of each pixel as a function of the pixel center to the closest edge distance.

Transformations are performed using matrix multiplication. The Vector Matrix Multiplier multiplies a 2 x 1 Vector (x,y) by a 3 x 3 homogeneous Matrix to produce a new 2 x 1 Vector (Tx,Ty). The following computation is required to calculate texel coordinates from screen coordinates:

$$T_x = \frac{t00 \cdot x + t01 \cdot y + t02}{t20 \cdot x + t21 \cdot y + t22} \qquad T_y = \frac{t10 \cdot x + t11 \cdot y + t12}{t20 \cdot x + t21 \cdot y + t22} \tag{25}$$

8.5.5.3.6 Texture Map Unit

The texture map unit produces texels that are sent to the fragment processing core. It is fed with texture's attributes (base address, dimensions, color format) and the required coordinates. The texture map unit performs some internal processing and outputs the corresponding texel. Generating a texture element requires a series of operations like wrapping (clamp, mirror, and repeat), reading corresponding color values from memory, converting the color values to RGBA8888 format, and performing filtering if necessary.

8.5.5.3.7 Fragment Processing Core

The fragment processing core is the main processing unit of the GPU's architecture. It is a 64-bit VLIW processor which performs computations on the fragments coming from the rasterizer unit and on the texels coming from the texture map unit and calculates the final color to a fragment. The core is programmable through binary executables called Fragment Shaders.

8.5.5.3.8 Render Output Unit

The render output unit (ROP) is the last stage of the graphics pipeline. The fragment processing core feeds the ROP with the pixel's coordinates and color value. Before the color value is written to the memory, the color is converted to the frambuffer's format.

If H/W Blender is enabled, the ROP reads pixels from the fragment processing core (source) and pixels from the framebuffer (destination) to perform blending. Blending requires a series of calculations between the source (foreground) and destination (background) color fragments to produce the final color, which is written back to memory. The following equations are used for the final color:

$$F_c = S_c \cdot S_f + D_c \cdot D_f \qquad F_a = S_a \cdot S_f + D_a \cdot D_f \tag{26}$$

- F_c: Final Color
- S_c: Source Color
- S_f: Source Blend Factor (multiplier)

- D_c: Destination Color
- D_f: Destination Blend Factor (multiplier)
- F_a: Final Alpha
- S_a: Source Alpha
- D_a: Destination Alpha

The Color and Alpha values range from 0 to 1, therefore each calculation result is also clamped to the same range. The available Blend Factors and the resulting RGBA values are listed in the table below. The effect of the blending modes is shown in the figure below.

Table 8-334 Blend factors

Blending Mode Name	Blend Factors (S_c or D_f)	RGBA Value	Definition
DSBF_ZERO	0	0,0,0,0	
DSBF_ONE	1	1,1,1,1	
DSBF_SRCCOLOR	S_c	R_{src}, G_{src}, B_{src}, A_{src}	• S_c: Source Color • R_{src}: Source Red value • G_{src}: Source Green value • B_{src}: Source Blue value • A_{src}: Source Alpha value
DSBF_INVSRCCOLOR	$(1 – S_c)$	$1 – R_{src}$, $1 – G_{src}$, $1–B_{src}$, $1 – A_{src}$	
DSBF_SRCALPHA	S_a	A_{src}, A_{src}, A_{src}, A_{src}	• S_a: Source Alpha • A_{src}: Source Alpha value
DSBF_INVSRCALPHA	$(1 – S_a)$	$1 – A_{src}$, $1 – A_{src}$, $1 – A_{src}$, $1 – A_{src}$	
DSBF_DESTALPHA	D_a	A_{dst}, A_{dst}, A_{dst}, A_{dst}	• D_a: Destination Alpha • A_{dst}: Destination Alpha value
DSBF_INVDESTALPHA	$(1 – D_a)$	$1 – A_{dst}$, $1 – A_{dst}$, $1 – A_{dst}$, $1 – A_{dst}$	
DSBF_DESTCOLOR	D_c	R_{dst}, G_{dst}, B_{dst}, A_{dst}	• D_c: Destination Color • R_{dst}: Destination Red value • G_{dst}: Destination Green value • B_{dst}: Destination Blue value • A_{dst}: Destination Alpha value
DSBF_INVDESTCOLOR	$(1 – D_c)$	$1 – R_{dst}$, $1 – G_{dst}$, $1 – B_{dst}$, $1 – A_{dst}$	
DSBF_CONSTCOLOR	C_c	R_{const}, G_{const}, B_{const}, A_{const}	• C_c: Constant Color • C_a: Constant Alpha • R_{const}: Constant Red value • G_{const}: Constant Green value • B_{const}: Constant Blue value • A_{const}: Constant Alpha value
DSBF_CONSTALPHA	C_a	A_{const}, A_{const}, A_{const}, A_{const}	
DSBF_UNKNOWN	0	0,0,0,0	

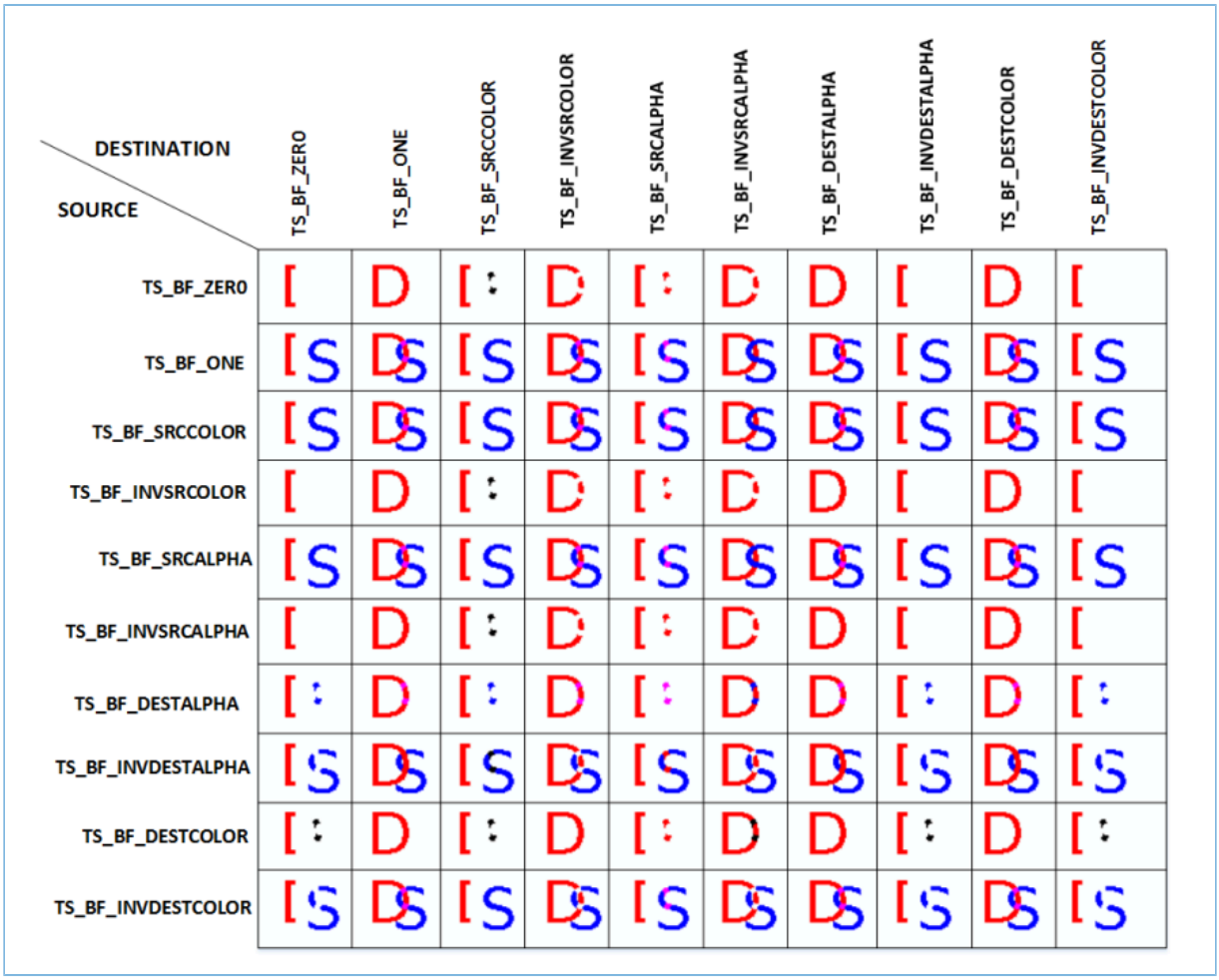

Figure 8-102 Blending modes

8.5.5.3.9 Framebuffer Compression

Framebuffer compression operates in screen blocks (4 x 4 pixel blocks) and achieves TSC^{TM}4, TSC^{TM}6 and TSC^{TM}6a lossy, fixed-ratio compression depending on the configuration.

- TSC^{TM}4 is a 6:1 compression (4bpp).
- TSC^{TM}6 is a 4:1 compression (6bpp).
- TSC^{TM}6a is a 4:1 compression (6bpp) with alpha channel.

Compression is performed at run time using minimal hardware. Pixel data can be stored in the framebuffer in compressed form and decompressed in the display control (DC) module. The output of the TSC^{TM}4 compression is 64 bits per 4 x 4 block of pixels, and the output of the TSC^{TM}6 compression is 96 bits per 4 x 4 block of pixels.

8.5.5.3.10 Color Modes

The GPU supports multiple color formats. The supported color formats are the following:

- RGBX8888 32-bit

Table 8-335 RGBX8888 32-bit

Pixel 0																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X7	X6	X5	X4	X3	X2	X1	X0	B7	B6	B5	B4	B3	B2	B1	B0	G7	G6	G5	G4	G3	G2	G1	G1	R7	R6	R5	R4	R3	R2	R1	R0
Byte 3								Byte 2								Byte 1								Byte 0							

- RGBA8888 32-bit

Table 8-336 RGBA8888 32-bit

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

- XRGB8888 32-bit

Table 8-337 XRGB8888 32-bit

Pixel 0																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B7	B6	B5	B4	B3	B2	B1	B0	G7	G6	G5	G4	G3	G2	G1	G1	R7	R6	R5	R4	R3	R2	R1	R0	X7	X6	X5	X4	X3	X2	X1	X0
Byte 3								Byte 2								Byte 1								Byte 0							

- ARGB8888 32-bit

Table 8-338 ARGB8888 32-bit

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

- RGB5650 16-bit

Table 8-339 RGB5650 16-bit

Pixel 0															
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	G1	B4	B3	B2	B1	B0
Byte 1								Byte 0							

- RGBA5551 16-bit

Table 8-340 RGBA5551 16-bit

Pixel 0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R4	R3	R2	R1	R0	G4	G3	G2	G1	G1	B4	B3	B2	B1	B0	A0
Byte 1								Byte 0							

- RGBA4444 16-bit

Table 8-341 RGBA4444 16-bit

Pixel 0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R3	R2	R1	R0	G3	G2	G1	G1	B3	B2	B1	B0	A3	A2	A1	A0
Byte 1								Byte 0							

- RGBA0008 8-bit

Table 8-342 RGBA0008 8-bit

Pixel 0							
7	6	5	4	3	2	1	0
A7	A6	A5	A4	A3	A2	A1	A0
Byte 0							

- L8 8-bit

Table 8-343 L8 8-bit

Pixel 0							
7	6	5	4	3	2	1	0
L7	L6	L5	L4	L3	L2	L1	L0
Byte 0							

- L1 big-endian 1-bit

Table 8-344 L1 big-endian 1-bit

Pixel 0	**Pixel 1**	**Pixel 2**	**Pixel 3**	**Pixel 4**	**Pixel 5**	**Pixel 6**	**Pixel 7**
7	6	5	4	3	2	1	0
L0	L0	L0	L0	L0	L0	L0	L0
Byte 0							

- A1 big-endian 1-bit

Table 8-345 A1 big-endian 1-bit

Pixel 0	**Pixel 1**	**Pixel 2**	**Pixel 3**	**Pixel 4**	**Pixel 5**	**Pixel 6**	**Pixel 7**
7	6	5	4	3	2	1	0
A0	A0	A0	A0	A0	A0	A0	A0
Byte 0							

- UYVY 32-bit 2-pixel

Table 8-346 UYVY 32-bit 2-pixel

Pixel 1								**Pixel1&0**								**Pixel 0**								**Pixel1&0**							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	V7	V6	V5	V4	V3	V2	V1	V0	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y1	U7	U6	U5	U4	U3	U2	U1	U0
Byte 3								Byte 2								Byte 1								Byte 0							

- ABGR8888 32-bit

Table 8-347 ABGR8888 32-bit

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

- BGRA 32-bit

Table 8-348 BGRA 32-bit

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

- BGRX 32-bit

Table 8-349 BGRX 32-bit

Pixel 0																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X7	X6	X5	X4	X3	X2	X1	X0	R7	R6	R5	R4	R3	R2	R1	R0	G7	G6	G5	G4	G3	G2	G1	G1	B7	B6	B5	B4	B3	B2	B1	B0
Byte 3								Byte 2								Byte 1								Byte 0							

- TSC4 16-pixel/64-bit

- TSC6 16-pixel/96-bit
- TSC6A 16-pixel with Alpha/96-bit
- A1LE little-endian 1-bit

Table 8-350 A1LE little-endian 1-bit

Pixel 7	Pixel 6	Pixel 5	Pixel 4	Pixel 3	Pixel 2	Pixel 1	Pixel 0
7	6	5	4	3	2	1	0
A0	A0	A0	A0	A0	A0	A0	A0
Byte 0							

- A2LE little-endian 2-bit

Table 8-351 A2LE little-endian 2-bit

Pixel 3		Pixel 2		Pixel 1		Pixel 0	
7	6	5	4	3	2	1	0
A1	A0	A1	A0	A1	A0	A1	A0
Byte 0							

- A4LE little-endian 4-bit

Table 8-352 A4LE little-endian 4-bit

Pixel 1				Pixel 0			
7	6	5	4	3	2	1	0
A3	A2	A1	A0	A3	A2	A1	A0
Byte 0							

- L1LE little-endian 1-bit

Table 8-353 L1LE little-endian 1-bit

Pixel 7	Pixel 6	Pixel 5	Pixel 4	Pixel 3	Pixel 2	Pixel 1	Pixel 0
7	6	5	4	3	2	1	0
L0	L0	L0	L0	L0	L0	L0	L0
Byte 0							

- L2LE little-endian 2-bit

Table 8-354 L2LE little-endian 2-bit

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

Pixel 3		Pixel 2		Pixel 1		Pixel 0	
L1	L0	L1	L0	L1	L0	L1	L0
Byte 0							

- L4LE little-endian 4-bit

Table 8-355 L4LE little-endian 4-bit

Pixel 1				Pixel 0			
7	6	5	4	3	2	1	0
L3	L2	L1	L0	L3	L2	L1	L0
Byte 0							

- A2 big-endian 2-bit

Table 8-356 A2 big-endian 2-bit

Pixel 0		Pixel 1		Pixel 2		Pixel 3	
7	6	5	4	3	2	1	0
A1	A0	A1	A0	A1	A0	A1	A0
Byte 0							

- L2 big-endian 2-bit

Table 8-357 L2 big-endian 2-bit

Pixel 0		Pixel 1		Pixel 2		Pixel 3	
7	6	5	4	3	2	1	0
L1	L0	L1	L0	L1	L0	L1	L0
Byte 0							

- A4 big-endian 4-bit

Table 8-358 A4 big-endian 4-bit

Pixel 0				Pixel 1			
7	6	5	4	3	2	1	0
A3	A2	A1	A0	A3	A2	A1	A0
Byte 0							

- L4 big-endian 4-bit

Table 8-359 L4 big-endian 4-bit

Pixel 0				**Pixel 1**			
7	6	5	4	3	2	1	0
L3	L2	L1	L0	L3	L2	L1	L0
Byte 0							

- RGBA3320 8-bit

Table 8-360 RGBA3320 8-bit

Pixel 0							
7	6	5	4	3	2	1	0
R2	R1	R0	G2	G1	G0	B1	B0
Byte 0							

- BGR24 24-bit

Table 8-361 BGR24 24-bit

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

- RGB24 24-bit

Table 8-362 RGB24 24-bit

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

8.5.5.3.11 Color Expansion

The internal format is always on RGBA8888 32-bit format. Therefore, lower-order color formats are expanded to 8 bits per color channel. This is achieved by high-order bit replication. For example, a 5-bit color format is constructed as follows:

$$C[7:0] = \{C[4:0]; C[4:2]\} \tag{27}$$

8.5.5.3.12 Instruction Set Architecture

The microarchitecture of the GPU for fragment processing is a four-issue VLIW core. Each VLIW instruction is 64 bits' long and consists of four sub-instructions (corresponding to sub-operations). The sub-operations for the GPU are: i)

instructions for Alpha channel arithmetic operations (SUBOP0); ii) instructions for RGB vector arithmetic operations (SUBOP1); iii) memory addressing instructions for reading and writing to memory (SUBOP2); and iv) branching instructions based on comparisons between RGB or Alpha values (SUBOP3).

Each VLIW bundle is designed to perform six arithmetic operations per cycle (simultaneously). If more than one VLIW bundle is enabled, the addressing sub-operations (SUBOP2) are issued once for all the enabled bundles while the remaining sub-operations (SUBOP0, SUBOP1, and SUBOP3) are issued once for each VLIW bundle.

8.5.5.4 Registers

8.5.5.4.1 TEX0_BASE

- **Name:** Base Address Register 0
- **Description:** This register specifies the base address of drawing surface 0.
- **Base Address:** 0x40028000
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 8-363 Base Address Register 0

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the base address of drawing surface 0.

8.5.5.4.2 TEX0_FSTRIDE

- **Name:** Image Stride Register 0
- **Description:** This register specifies image 0 mode and stride.
- **Base Address:** 0x40028000
- **Offset:** 0x4
- **Reset Value:** 0x00000000

Table 8-364 Image Stride Register 0

Bits	Field Name	RW	Reset	Description
31		RW	0x0	If set for destination texture, color dithering is enabled. (Applicable to output formats: RGBA5650, RGBA5551, and RGBA4444).
30:24		RW	0x0	Specifies the image format. • 0x00: RGBX8888 • 0x01: RGBA8888 • 0x02: XRGB8888 • 0x03: ARGB8888

Bits	Field Name	RW	Reset	Description
				• 0x04: RGBA5650 • 0x05: RGBA555 • 0x06: RGBA44441 • 0x08: RGBA0008 (only available as input format) • 0x09: L8 • 0x0B: L1 (only available as input format) • 0x0C: A1 (only available as input format) • 0x0D: UYVY (only available as input format) • 0x0E: ABGR8888 (only available as input format) • 0x10: BGRA (only available as input format) • 0x11: BGRX (only available as input format) • 0x12: TSC4 • 0x16: TSC6 • 0x17: TSC6A • 0x27: A1LE (only available as input format) • 0x28: A2LE (only available as input format) • 0x29: A4LE (only available as input format) • 0x2A: L1LE (only available as input format) • 0x2B: L2LE (only available as input format) • 0x2C: L4LE (only available as input format) • 0x30: A2 (only available as input format) • 0x31: L2 (only available as input format) • 0x34: A4 (only available as input format) • 0x35: L4 (only available as input format) • 0x38: RGBA3320 (only available as input format) • 0x39: BGR24 • 0x3C: RGB24
23:16		RW	0x0	Specifies the image mode. • 0x00: Point sampling • 0x01: Bilinear filtering
15:0		RW	0x0	Specifies the image stride distance in bytes from one scanline to another (signed).

8.5.5.4.3 TEX0_RESXY

- **Name:** Image Size Register 0
- **Description:** This register specifies the size of image 0 resolution.
- **Base Address:** 0x40028000
- **Offset:** 0x8
- **Reset Value:** 0x00000000

Table 8-365 Image Size Register 0

Bits	Field Name	RW	Reset	Description
31:16		RW	0x0	Specifies the size of resolution Y.
15:0		RW	0x0	Specifies the size of resolution X.

8.5.5.4.4 TEX1_BASE

- **Name:** Base Address Register 1
- **Description:** This register specifies the base address of drawing surface 1.
- **Base Address:** 0x40028000
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 8-366 Base Address Register 1

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the base address of drawing surface 1.

8.5.5.4.5 TEX1_FSTRIDE

- **Name:** Image Stride Register 1
- **Description:** This register specifies image 1 mode and stride.
- **Base Address:** 0x40028000
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 8-367 Image Stride Register 1

Bits	Field Name	RW	Reset	Description
31		RW	0x0	If set for destination texture, enables color dithering. (Applicable to output formats: RGBA5650, RGBA5551, and RGBA4444).

Bits	Field Name	RW	Reset	Description
30:24		RW	0x0	Specifies the image format. • 0x00: RGBX8888 • 0x01: RGBA8888 • 0x02: XRGB8888 • 0x03: ARGB8888 • 0x04: RGBA5650 • 0x05: RGBA5551 • 0x06: RGBA4444 • 0x08: RGBA0008 (only available as input format) • 0x09: L8 • 0x0B: L1 (only available as input format) • 0x0C: A1 (only available as input format) • 0x0D: UYVY (only available as input format) • 0x0E: ABGR8888 (only available as input format) • 0x10: BGRA (only available as input format) • 0x11: BGRX (only available as input format) • 0x12: TSC4 • 0x16: TSC6 • 0x17: TSC6A • 0x27: A1LE (only available as input format) • 0x28: A2LE (only available as input format) • 0x29: A4LE (only available as input format) • 0x2A: L1LE (only available as input format) • 0x2B: L2LE (only available as input format) • 0x2C: L4LE (only available as input format) • 0x30: A2 (only available as input format) • 0x31: L2 (only available as input format) • 0x34: A4 (only available as input format) • 0x35: L4 (only available as input format) • 0x38: RGBA3320 (only available as input format) • 0x39: BGR24 • 0x3C: RGB24
23:16		RW	0x0	Specifies the image mode.

Bits	Field Name	RW	Reset	Description
				• 0x00: Point sampling • 0x01: Bilinear filtering
15:0		RW	0x0	Specifies the image stride distance in bytes from one scanline to another (signed).

8.5.5.4.6 TEX1_RESXY

- **Name:** Image Size Register 1
- **Description:** This register specifies the size of image 1 resolution.
- **Base Address:** 0x40028000
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 8-368 Image Size Register 1

Bits	Field Name	RW	Reset	Description
31:16		RW	0x0	Specifies the size of resolution Y.
15:0		RW	0x0	Specifies the size of resolution X.

8.5.5.4.7 TEX_COLOR

- **Name:** Color Register
- **Description:** This register specifies texture map's default color.
- **Base Address:** 0x40028000
- **Offset:** 0x1C
- **Reset Value:** 0x00000000

Table 8-369 Color Register

Bits	Field Name	RW	Reset	Description
31:24		RW	0x0	Specifies the Alpha (A) value.
23:16		RW	0x0	Specifies the Blue (B) value.
15:8		RW	0x0	Specifies the Green (G) value.
7:0		RW	0x0	Specifies the Red (R) value.

8.5.5.4.8 TEX2_BASE

- **Name:** Base Address Register 2
- **Description:** This register specifies the base address of drawing surface 2.

- **Base Address:** 0x40028000
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 8-370 Base Address Register 2

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the base address of drawing surface 2.

8.5.5.4.9 TEX2_FSTRIDE

- **Name:** Image Stride Register 2
- **Description:** This register specifies image 2 mode and stride.
- **Base Address:** 0x40028000
- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 8-371 Image Stride Register 2

Bits	Field Name	RW	Reset	Description
31		RW	0x0	If set for destination texture, color dithering is enabled. (Applicable to output formats: RGBA5650, RGBA5551, and RGBA4444).
30:24		RW	0x0	Specifies the image format. • 0x00: RGBX8888 • 0x01: RGBA8888 • 0x02: XRGB8888 • 0x03: ARGB8888 • 0x04: RGBA5650 • 0x05: RGBA5551 • 0x06: RGBA4444 • 0x08: RGBA0008 (only available as input format) • 0x09: L8 • 0x0B: L1 (only available as input format) • 0x0C: A1 (only available as input format) • 0x0D: UYVY (only available as input format) • 0x0E: ABGR8888 (only available as input format) • 0x10: BGRA (only available as input format) • 0x11: BGRX (only available as input format)

Bits	Field Name	RW	Reset	Description
				• 0x12: TSC4 • 0x16: TSC6 • 0x17: TSC6A • 0x27: A1LE (only available as input format) • 0x28: A2LE (only available as input format) • 0x29: A4LE (only available as input format) • 0x2A: L1LE (only available as input format) • 0x2B: L2LE (only available as input format) • 0x2C: L4LE (only available as input format) • 0x30: A2 (only available as input format) • 0x31: L2 (only available as input format) • 0x34: A4 (only available as input format) • 0x35: L4 (only available as input format) • 0x38: RGBA3320 (only available as input format) • 0x39: BGR24 • 0x3C: RGB24
23:16		RW	0x0	Specifies the image mode. • 0x00: Point sampling • 0x01: Bilinear filtering
15:0		RW	0x0	Specifies the image stride distance in bytes from one scanline to another (signed).

8.5.5.4.10 TEX2_RESXY

- **Name:** Image Size Register 2
- **Description:** This register specifies the size of image 2 resolution.
- **Base Address:** 0x40028000
- **Offset:** 0x28
- **Reset Value:** 0x00000000

Table 8-372 Image Size Register 2

Bits	Field Name	RW	Reset	Description
31:16		RW	0x0	Specifies the size of resolution Y.
15:0		RW	0x0	Specifies the size of resolution X.

8.5.5.4.11 TEX3_BASE

- **Name:** Base Address Register 3
- **Description:** This register specifies the base address of drawing surface 3.
- **Base Address:** 0x40028000
- **Offset:** 0x30
- **Reset Value:** 0x00000000

Table 8-373 Base Address Register 3

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the base address of drawing surface 3.

8.5.5.4.12 TEX3_FSTRIDE

- **Name:** Image Stride Register 3
- **Description:** This register specifies image 3 mode and stride.
- **Base Address:** 0x40028000
- **Offset:** 0x34
- **Reset Value:** 0x00000000

Table 8-374 Image Stride Register 3

Bits	Field Name	RW	Reset	Description
31		RW	0x0	If set for destination texture, color dithering is enabled. (Applicable to output formats: RGBA5650, RGBA5551, and RGBA4444).
30:24		RW	0x0	Specifies the image format. • 0x00: RGBX8888 • 0x01: RGBA8888 • 0x02: XRGB8888 • 0x03: ARGB8888 • 0x04: RGBA5650 • 0x05: RGBA5551 • 0x06: RGBA4444 • 0x08: RGBA0008 (only available as input format) • 0x09: L8 • 0x0B: L1 (only available as input format) • 0x0C: A1 (only available as input format)

Bits	Field Name	RW	Reset	Description
				• 0x0D: UYVY (only available as input format) • 0x0E: ABGR8888 (only available as input format) • 0x10: BGRA (only available as input format) • 0x11: BGRX (only available as input format) • 0x12: TSC4 • 0x16: TSC6 • 0x17: TSC6A • 0x27: A1LE (only available as input format) • 0x28: A2LE (only available as input format) • 0x29: A4LE (only available as input format) • 0x2A: L1LE (only available as input format) • 0x2B: L2LE (only available as input format) • 0x2C: L4LE (only available as input format) • 0x30: A2 (only available as input format) • 0x31: L2 (only available as input format) • 0x34: A4 (only available as input format) • 0x35: L4 (only available as input format) • 0x38: RGBA3320 (only available as input format) • 0x39: BGR24 • 0x3C: RGB24
23:16		RW	0x0	Specifies the image mode. • 0x00: Point sampling • 0x01: Bilinear filtering
15:0		RW	0x0	Specifies the image stride distance in bytes from one scanline to another (signed).

8.5.5.4.13 TEX3_RESXY

- **Name:** Image Size Register 3
- **Description:** This register specifies the size of image 3 resolution.
- **Base Address:** 0x40028000
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 8-375 Image Size Register 3

Bits	Field Name	RW	Reset	Description
31:16		RW	0x0	Specifies the size of resolution Y.
15:0		RW	0x0	Specifies the size of resolution X.

8.5.5.4.14 BREAKPOINT

- **Name:** Break Point Register
- **Description:** This register specifies the break point.
- **Base Address:** 0x40028000
- **Offset:** 0x80
- **Reset Value:** 0x00000000

Table 8-376 Break Point Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	

8.5.5.4.15 BREAKPOINT_MASK

- **Name:** Break Point Mask Register
- **Description:** This register specifies the break point mask.
- **Base Address:** 0x40028000
- **Offset:** 0x8C
- **Reset Value:** 0x00000000

Table 8-377 Break Point Mask Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	

8.5.5.4.16 CG_CTRL

- **Name:** Clock Gating Control Register
- **Description:** This register controls the clock gating.
- **Base Address:** 0x40028000
- **Offset:** 0x94
- **Reset Value:** 0x00000000

Table 8-378 Clock Gating Control Register

Bits	Field Name	RW	Reset	Description
31		RW	0x0	Disable clock gating for all modules.
30:28	RSVD	R		Reserved bits
27		RW	0x0	Disable clock gating for Core 0 Instruction Memory.
26:24	RSVD	R		Reserved bits
23		RW	0x0	Disable clock gating for Core 0.
22:20	RSVD	R		Reserved bits
19		RW	0x0	Disable clock gating for Core 0 Scheduler.
18:16	RSVD	R		Reserved bits
15		RW	0x0	Disable clock gating for Core 0 Register File.
14:12	RSVD	R		Reserved bits
11		RW	0x0	Disable clock gating for Core 0 Pipeline.
10:8	RSVD	R		Reserved bits
7		RW	0x0	Disable clock gating for the BusInterface (memory system).
6:4	RSVD	R		Reserved bits
3		RW	0x0	Disable clock gating for the Render Output Unit of Core 0.
2	RSVD	R		Reserved bit
1		RW	0x0	Disable clock gating for the NemaP Configuration Register File.
0		RW	0x0	Disable clock gating for the Command List Processor.

8.5.5.4.17 DIRTYMIN

- **Name:** Dirty Region Min Register
- **Description:** This register reads the Dirty_Min value and resets dirty region to resolution size on write function.
- **Base Address:** 0x40028000
- **Offset:** 0x98
- **Reset Value:** 0xFFFFFFFF

Table 8-379 Dirty Region Min Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0xFFFFFFFF	Write any value to reset dirty region to resolution size. Read the Dirty_Min value.

8.5.5.4.18 DIRTYMAX

- **Name:** Dirty Region Max Register
- **Description:** This register reads the Dirty_Max value.
- **Base Address:** 0x40028000
- **Offset:** 0x9C
- **Reset Value:** 0x00000000

Table 8-380 Dirty Region Max Register

Bits	Field Name	RW	Reset	Description
31:0		R	0x0	Read the Dirty_Max value.

8.5.5.4.19 IMEM_ADDR

- **Name:** Load Shader Instruction Memory Address Register
- **Description:** This register specifies the load shader instruction memory address.
- **Base Address:** 0x40028000
- **Offset:** 0xC4
- **Reset Value:** 0x00000000

Table 8-381 Load Shader Instruction Memory Address Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0		RW	0x0	Specifies the load shader instruction memory address.

8.5.5.4.20 IMEM_DATA_LOW

- **Name:** Lower Bit Data Register
- **Description:** This register specifies the lower bits (31: 0) of the 64-bit load shader instruction memory data.
- **Base Address:** 0x40028000
- **Offset:** 0xC8
- **Reset Value:** 0x00000000

Table 8-382 Lower Bit Data Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the lower bits (31:0) of the load shader instruction memory data.

8.5.5.4.21 IMEM_DATA_HIGH

- **Name:** Higher Bit Data Register

- **Description:** This register specifies the higher bits (63-32) of the 64-bit load shader instruction memory data.
- **Base Address:** 0x40028000
- **Offset:** 0xCC
- **Reset Value:** 0x00000000

Table 8-383 Higher Bit Data Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the higher bits (63-32) of the load shader instruction memory data.

8.5.5.4.22 FLUSH_CTRL

- **Name:** System Flush Register
- **Description:** This register controls the system flush.
- **Base Address:** 0x40028000
- **Offset:** 0xE4
- **Reset Value:** 0x00000007

Table 8-384 System Flush Register

Bits	Field Name	RW	Reset	Description
31:14	RSVD	R		Reserved bits
13		RW	0x0	When set to 1, the write caches are flushed.
12	RSVD	R		Reserved bit
11		RW	0x0	If set to 1, the write caches are flushed when the host triggers the rasterizer register (NEMA_DRAW_CMD_NOHOLD).
10:8	RSVD	R		Reserved bits
7		RW	0x0	If set to 1, the write caches are flushed when the Command List Processor triggers the rasterizer register (NEMA_DRAW_CMD_NOHOLD).
6		RW	0x0	When set to 1, the read caches are flushed.
5	RSVD	R		Reserved bit
4		RW	0x0	If set to 1, the read caches are flushed when the host triggers the rasterizer register (NEMA_DRAW_CMD_NOHOLD)
3:1	RSVD	R		Reserved bits
0		RW	0x1	If set to 1, the read caches are flushed when the Command List Processor triggers the rasterizer register (NEMA_DRAW_CMD_NOHOLD).

8.5.5.4.23 CMD_STAT

- **Name:** Command Status Register
- **Description:** This register shows the command status.
- **Base Address:** 0x40028000
- **Offset:** 0xE8
- **Reset Value:** 0x00000000

Table 8-385 Command Status Register

Bits	Field Name	RW	Reset	Description
31		RW	0x0	Indicates if the Command List Processor is busy.
30		RW	0x0	Indicates if there are data available in System Memory.
29		RW	0x0	Indicates if there are available data to read from FIFO.
28		RW	0x0	Shows many cells are still available in FIFO.
27:18	RSVD	R		Reserved bits
17:13		RW	0x0	Shows many cells are still available in FIFO.
12	RSVD	R		Reserved bit
11:9		RW	0x0	Shows the address of the stack pointer.
8:7	RSVD	R		Reserved bits
6:4		RW	0x0	Indicates the state of CMDList internal FSM.
3:1	RSVD	R		Reserved bits
0		RW	0x0	Indicates the state of the FSM responsible for fetching.

8.5.5.4.24 RING_BUF_STORE

- **Name:** Ring Buffer Register
- **Description:** This register stores the Ring Buffer's last written address.
- **Base Address:** 0x40028000
- **Offset:** 0xEC
- **Reset Value:** 0x00000000

Table 8-386 Ring Buffer Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Stores the last written address of the ring buffer.

8.5.5.4.25 CMD_ADDR

- **Name:** Command List Base Address Register
- **Description:** This register specifies the command list base address.

- **Base Address:** 0x40028000
- **Offset:** 0xF0
- **Reset Value:** 0x00000000

Table 8-387 Command List Base Address Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the base address of every new command list ready to be executed.

8.5.5.4.26 CMD_SIZE

- **Name:** Command List Size Register
- **Description:** This register specifies the command list size.
- **Base Address:** 0x40028000
- **Offset:** 0xF4
- **Reset Value:** 0x00000000

Table 8-388 Command List Size Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the length of the command list ready to be executed.

8.5.5.4.27 INT_CTRL

- **Name:** Interrupt Control Register
- **Description:** This register controls the interrupt.
- **Base Address:** 0x40028000
- **Offset:** 0xF8
- **Reset Value:** 0x00000000

Table 8-389 Interrupt Control Register

Bits	Field Name	RW	Reset	Description
31:30		RW	0x0	Specifies the core_clk_dvfs output.
29:28	RSVD	R		Reserved bits
27		RW	0x0	When set to 1, the throttle signal will be ignored.
26:4	RSVD	R		Reserved bits
3		RW	0x0	When set to 1, the interrupt in the NEMA_IRQ_ID register is cleared.
2		RW	0x0	When set to 1, an interrupt is signaled at the end of a drawing command.
1		RW	0x0	When set to 1, an interrupt is signaled at the end of a command list.

Bits	Field Name	RW	Reset	Description
0		RW	0x0	• 1: IRQ signal is active low. • 0: IRQ signal is active high.

8.5.5.4.28 STAT

- **Name:** GPU Status Register
- **Description:** This register shows the GPU status.
- **Base Address:** 0x40028000
- **Offset:** 0xFC
- **Reset Value:** 0x00000000

Table 8-390 GPU Status Register

Bits	Field Name	RW	Reset	Description
31		RW	0x0	Indicates if the system is busy.
30		RW	0x0	Indicates if the bus interface unit is busy.
29		RW	0x0	Indicates if the command list bus is busy.
28		RW	0x0	Indicates if the Command List Processor is busy.
27:24		RW	0x0	Indicate if the rasterizer unit is busy.
23:13	RSVD	R		Reserved bits
12		RW	0x0	Indicates if the Core 0 render output unit is busy.
11:9	RSVD	R		Reserved bits
8		RW	0x0	Indicates if the Core 0 texture map unit is busy.
7:5	RSVD	R		Reserved bits
4		RW	0x0	Indicates if the Core 0 graphics pipeline is busy.
3:1	RSVD	R		Reserved bits
0		RW	0x0	Indicates if the Core 0 is busy.

8.5.5.4.29 DRAW_CMD_NOHOLD

- **Name:** Rasterizer Command Register
- **Description:** This register controls the rasterizer.
- **Base Address:** 0x40028000
- **Offset:** 0x100
- **Reset Value:** 0x00000000

Table 8-391 Rasterizer Command Register

Bits	Field Name	RW	Reset	Description
31	RSVD	R		Reserved bit
30		RW	0x0	When set to 1, tile mode is enabled.
29		RW	0x0	When set to 1, positive culling (counter-clockwise) is enabled.
28		RW	0x0	When set to 1, negative culling (clockwise) is enabled.
27		RW	0x0	When set to 1, color gradient is enabled.
26		RW	0x0	When set to 1, antialiasing feature is enabled for edge 0.
25		RW	0x0	When set to 1, antialiasing feature is enabled for edge 1.
24		RW	0x0	When set to 1, antialiasing feature is enabled for edge 2.
23		RW	0x0	When set to 1, antialiasing feature is enabled for edge 3.
22:3	RSVD	R		Reserved bits
2:0		RW	0x0	Draw commands: • 0x00: Draw pixels. • 0x01: Draw lines. • 0x02: Draw rectangles. • 0x03: Draw triangles. • 0x04: Draw quadrilaterals.

8.5.5.4.30 DRAW_STARTXY

- **Name:** Draw Start XY Register
- **Description:** This register specifies the start XY of drawing.
- **Base Address:** 0x40028000
- **Offset:** 0x104
- **Reset Value:** 0x00000000

Table 8-392 Draw Start XY Register

Bits	Field Name	RW	Reset	Description
31:16		RW	0x0	Specifies the Y coordinate (integer value) of vertex 0.
15:0		RW	0x0	Specifies the X coordinate (integer value) of vertex 0.

8.5.5.4.31 DRAW_ENDXY

- **Name:** Draw End XY Register
- **Description:** This register specifies the end XY of drawing.

- **Base Address:** 0x40028000
- **Offset:** 0x108
- **Reset Value:** 0x00000000

Table 8-393 Draw End XY Register

Bits	Field Name	RW	Reset	Description
31:16		RW	0x0	Specifies the Y coordinate (integer value) of vertex 1.
15:0		RW	0x0	Specifies the X coordinate (integer value) of vertex 1.

8.5.5.4.32 CLIPMIN

- **Name:** Clip Min Register
- **Description:** This register specifies clipping rectangle upper left vertex.
- **Base Address:** 0x40028000
- **Offset:** 0x110
- **Reset Value:** 0x00000000

Table 8-394 Clip Min Register

Bits	Field Name	RW	Reset	Description
31:16		RW	0x0	Defines the upper left Y coordinate.
15:0		RW	0x0	Defines the upper left X coordinate.

8.5.5.4.33 CLIPMAX

- **Name:** Clip Max Register
- **Description:** This register specifies clipping rectangle bottom right vertex.
- **Base Address:** 0x40028000
- **Offset:** 0x114
- **Reset Value:** 0x00000000

Table 8-395 Clip Max Register

Bits	Field Name	RW	Reset	Description
31:16		RW	0x0	Defines the bottom right Y coordinate.
15:0		RW	0x0	Defines the bottom right X coordinate.

8.5.5.4.34 MAT_MCL

- **Name:** Matrix Multiplication Control Register

- **Description:** This register controls the rasterizer matrix multiplication.
- **Base Address:** 0x40028000
- **Offset:** 0x118
- **Reset Value:** 0x00000000

Table 8-396 Matrix Multiplication Control Register

Bits	Field Name	RW	Reset	Description
31:30	RSVD	R		Reserved bits
29		RW	0x0	When set to 1, the 0.5 value is added to the X and Y coordinates.
28		RW	0x0	When set to 1, the matrix multiplications are bypassed.
27		RW	0x0	When set to 0, the result is in fixed point value (default).
26		RW	0x0	When set to 1, the result is in floating point value.
25:0	RSVD	R		Reserved bits

8.5.5.4.35 CODEPTR

- **Name:** Shader Code Register
- **Description:** This register specifies the pointer of the shader code.
- **Base Address:** 0x40028000
- **Offset:** 0x11C
- **Reset Value:** 0x00000000

Table 8-397 Shader Code Register

Bits	Field Name	RW	Reset	Description
31:16		RW	0x0	Specifies the pointer for the instruction that will be executed for the background pixels.
15:0		RW	0x0	Specifies the pointer for the instruction that will be executed for the foreground pixels.

8.5.5.4.36 DRAW_PT0_X

- **Name:** Vertex 0 Draw X Register
- **Description:** This register specifies the X coordinate of vertex 0 drawing primitive.
- **Base Address:** 0x40028000
- **Offset:** 0x120

- **Reset Value:** 0x00000000

Table 8-398 Vertex 0 Draw X Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the X coordinate of vertex 0 drawing primitive (16.16 fixed point).

8.5.5.4.37 DRAW_PT0_Y

- **Name:** Vertex 0 Draw Y Register
- **Description:** This register specifies the Y coordinate of vertex 0 drawing primitive.
- **Base Address:** 0x40028000
- **Offset:** 0x124
- **Reset Value:** 0x00000000

Table 8-399 Vertex 0 Draw Y Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the Y coordinate of vertex 0 drawing primitive (16.16 fixed point).

8.5.5.4.38 DRAW_COLOR

- **Name:** Drawing Color Register
- **Description:** This register specifies the drawing color for filling primitives.
- **Base Address:** 0x40028000
- **Offset:** 0x12C
- **Reset Value:** 0x00000000

Table 8-400 Drawing Color Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the drawing color for filling primitives.

8.5.5.4.39 DRAW_PT1_X

- **Name:** Vertex 1 Draw X Register
- **Description:** This register specifies the X coordinate of vertex 1 drawing primitive.
- **Base Address:** 0x40028000
- **Offset:** 0x130

- **Reset Value:** 0x00000000

Table 8-401 Vertex 1 Draw X Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the X coordinate of vertex 1 drawing primitive (16.16 fixed point).

8.5.5.4.40 DRAW_PT1_Y

- **Name:** Vertex 1 Draw Y Register
- **Description:** This register specifies the Y coordinate of vertex 1 drawing primitive.
- **Base Address:** 0x40028000
- **Offset:** 0x134
- **Reset Value:** 0x00000000

Table 8-402 Vertex 1 Draw Y Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the Y coordinate of vertex 1 drawing primitive (16.16 fixed point).

8.5.5.4.41 BYPASS_ADDR

- **Name:** Bypass Address Register
- **Description:** This register shows the address of bypass mode of rasterizer setup access.
- **Base Address:** 0x40028000
- **Offset:** 0x138
- **Reset Value:** 0x00000000

Table 8-403 Bypass Address Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Sets the address of bypass mode of rasterizer setup access.

8.5.5.4.42 BYPASS_DATA

- **Name:** Bypass Data Register
- **Description:** This register shows the data and enablement of bypass mode of rasterizer.
- **Base Address:** 0x40028000
- **Offset:** 0x13C

- **Reset Value:** 0x00000000

Table 8-404 Bypass Data Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Shows data and enablement of bypass mode of rasterizer.

8.5.5.4.43 DRAW_PT2_X

- **Name:** Vertex 2 Draw X Register
- **Description:** This register specifies the X coordinate of vertex 2 drawing primitive.
- **Base Address:** 0x40028000
- **Offset:** 0x140
- **Reset Value:** 0x00000000

Table 8-405 Vertex 2 Draw X Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the X coordinate of vertex 2 drawing primitive (16.16 fixed point).

8.5.5.4.44 DRAW_PT2_Y

- **Name:** Vertex 2 Draw Y Register
- **Description:** This register specifies the Y coordinate of vertex 2 drawing primitive.
- **Base Address:** 0x40028000
- **Offset:** 0x144
- **Reset Value:** 0x00000000

Table 8-406 Vertex 2 Draw Y Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the Y coordinate of vertex 2 drawing primitive (16.16 fixed point).

8.5.5.4.45 CMD_LIST_ID

- **Name:** Command List ID Register
- **Description:** This register specifies the last executed command list.
- **Base Address:** 0x40028000
- **Offset:** 0x148

- **Reset Value:** 0x00000000

Table 8-407 Command List ID Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the last executed command list.

8.5.5.4.46 DRAW_PT3_X

- **Name:** Vertex 3 Draw X Register
- **Description:** This register specifies the X coordinate of vertex 3 drawing primitive.
- **Base Address:** 0x40028000
- **Offset:** 0x150
- **Reset Value:** 0x00000000

Table 8-408 Vertex 3 Draw X Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the X coordinate of vertex 3 drawing primitive (16.16 fixed point).

8.5.5.4.47 DRAW_PT3_Y

- **Name:** Vertex 3 Draw Y Register
- **Description:** This register specifies the Y coordinate of vertex 3 drawing primitive.
- **Base Address:** 0x40028000
- **Offset:** 0x154
- **Reset Value:** 0x00000000

Table 8-409 Vertex 3 Draw Y Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the Y coordinate of vertex 3 drawing primitive (16.16 fixed point).

8.5.5.4.48 CLIPMIN1

- **Name:** Clip Min Register 1
- **Description:** This register specifies the second clipping rectangle upper left vertex.
- **Base Address:** 0x40028000
- **Offset:** 0x158

- **Reset Value:** 0x00000000

Table 8-410 Clip Min Register 1

Bits	Field Name	RW	Reset	Description
31:16		RW	0x0	Defines the upper left Y coordinate.
15:0		RW	0x0	Defines the upper left X coordinate.

8.5.5.4.49 CLIPMAX1

- **Name:** Clip Max Register 1
- **Description:** This register specifies the second clipping rectangle bottom right vertex.
- **Base Address:** 0x40028000
- **Offset:** 0x15C
- **Reset Value:** 0x7FFF7FFF

Table 8-411 Clip Max Register 1

Bits	Field Name	RW	Reset	Description
31:16		RW	0x7FFF	Defines the bottom right Y coordinate.
15:0		RW	0x7FFF	Defines the bottom right X coordinate.

8.5.5.4.50 MM00

- **Name:** (0,0) Matrix Register
- **Description:** This register specifies the (0,0) matrix floating point element.
- **Base Address:** 0x40028000
- **Offset:** 0x160
- **Reset Value:** 0x00000000

Table 8-412 (0,0) Matrix Register

Bits	Field Name	RW	Reset	Description
31:21	RSVD	R		Reserved bits
20:0		RW	0x0	Specifies the (0,0) element.

8.5.5.4.51 MM01

- **Name:** (0,1) Matrix Register
- **Description:** This register specifies the (0,1) matrix floating point element.
- **Base Address:** 0x40028000

- **Offset:** 0x164
- **Reset Value:** 0x00000000

Table 8-413 (0,1) Matrix Register

Bits	Field Name	RW	Reset	Description
31:21	RSVD	R		Reserved bits
20:0		RW	0x0	Specifies the (0,1) element.

8.5.5.4.52 MM02

- **Name:** (0,2) Matrix Register
- **Description:** This register specifies the (0,2) matrix floating point element.
- **Base Address:** 0x40028000
- **Offset:** 0x168
- **Reset Value:** 0x00000000

Table 8-414 (0,2) Matrix Register

Bits	Field Name	RW	Reset	Description
31:21	RSVD	R		Reserved bits
20:0		RW	0x0	Specifies the (0,2) element.

8.5.5.4.53 MM10

- **Name:** (1,0) Matrix Register
- **Description:** This register specifies the (1,0) matrix floating point element.
- **Base Address:** 0x40028000
- **Offset:** 0x16C
- **Reset Value:** 0x00000000

Table 8-415 (1,0) Matrix Register

Bits	Field Name	RW	Reset	Description
31:21	RSVD	R		Reserved bits
20:0		RW	0x0	Specifies the (1,0) element.

8.5.5.4.54 MM11

- **Name:** (1,1) Matrix Register
- **Description:** This register specifies the (1,1) matrix floating point element.

- **Base Address:** 0x40028000
- **Offset:** 0x170
- **Reset Value:** 0x00000000

Table 8-416 (1,1) Matrix Register

Bits	Field Name	RW	Reset	Description
31:21	RSVD	R		Reserved bits
20:0		RW	0x0	Specifies the (1,1) element.

8.5.5.4.55 MM12

- **Name:** (1,2) Matrix Register
- **Description:** This register specifies the (1,2) matrix floating point element.
- **Base Address:** 0x40028000
- **Offset:** 0x174
- **Reset Value:** 0x00000000

Table 8-417 (1,2) Matrix Register

Bits	Field Name	RW	Reset	Description
31:21	RSVD	R		Reserved bits
20:0		RW	0x0	Specifies the (1,2) element.

8.5.5.4.56 MM20

- **Name:** (2,0) Matrix Register
- **Description:** This register specifies the (2,0) matrix floating point element.
- **Base Address:** 0x40028000
- **Offset:** 0x178
- **Reset Value:** 0x00000000

Table 8-418 (2,0) Matrix Register

Bits	Field Name	RW	Reset	Description
31:21	RSVD	R		Reserved bits
20:0		RW	0x0	Specifies the (2,0) element.

8.5.5.4.57 MM21

- **Name:** (2,1) Matrix Register

- **Description:** This register specifies the (2,1) matrix floating point element.
- **Base Address:** 0x40028000
- **Offset:** 0x17C
- **Reset Value:** 0x00000000

Table 8-419 (2,1) Matrix Register

Bits	Field Name	RW	Reset	Description
31:21	RSVD	R		Reserved bits
20:0		RW	0x0	Specifies the (2,1) element.

8.5.5.4.58 MM22

- **Name:** (2,2) Matrix Register
- **Description:** This register specifies the (2,2) matrix floating point element.
- **Base Address:** 0x40028000
- **Offset:** 0x180
- **Reset Value:** 0x00000000

Table 8-420 (2,2) Matrix Register

Bits	Field Name	RW	Reset	Description
31:21	RSVD	R		Reserved bits
20:0		RW	0x0	Specifies the (2,2) element.

8.5.5.4.59 DEPTH_START_L

- **Name:** Integral Depth Value Low Register
- **Description:** This register specifies the integral part of lower 32-bit depth value of the STARTXY pixel.
- **Base Address:** 0x40028000
- **Offset:** 0x184
- **Reset Value:** 0x00000000

Table 8-421 Integral Depth Value Low Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the integral part of the depth value of the STARTXY pixel.

8.5.5.4.60 DEPTH_START_H

- **Name:** Integral Depth Value High Register
- **Description:** This register specifies the integral part of higher 32-bit depth value of the STARTXY pixel.
- **Base Address:** 0x40028000
- **Offset:** 0x188
- **Reset Value:** 0x00000000

Table 8-422 Integral Depth Value High Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the integral value of the depth value of the STARTXY pixel.

8.5.5.4.61 DEPTH_DX_L

- **Name:** Fractional X Depth Value Low Register
- **Description:** This register specifies the fractional part of lower 32-bit depth value for each step at X axis.
- **Base Address:** 0x40028000
- **Offset:** 0x18C
- **Reset Value:** 0x00000000

Table 8-423 Fractional X Depth Value Low Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the fractional part of the added depth value for each step at X axis.

8.5.5.4.62 DEPTH_DX_H

- **Name:** Integral X Depth Value High Register
- **Description:** This register specifies the integral part of higher 32-bit depth value for each step at X axis.
- **Base Address:** 0x40028000
- **Offset:** 0x190
- **Reset Value:** 0x00000000

Table 8-424 Integral X Depth Value High Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the integral part of the added depth value for each step at X axis.

8.5.5.4.63 DEPTH_DY_L

- **Name:** Fractional Y Depth Value Low Register
- **Description:** This register specifies the fractional part of lower 32-bit depth value for each step at Y axis.
- **Base Address:** 0x40028000
- **Offset:** 0x194
- **Reset Value:** 0x00000000

Table 8-425 Fractional Y Depth Value Low Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the fractional part of the added depth value for each step at Y axis.

8.5.5.4.64 DEPTH_DY_H

- **Name:** Integral Y Depth Value High Register
- **Description:** This register specifies the integral part of higher 32-bit depth value for each step at Y axis.
- **Base Address:** 0x40028000
- **Offset:** 0x198
- **Reset Value:** 0x00000000

Table 8-426 Integral Y Depth Value High Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the integral part of the added depth value for each step at Y axis.

8.5.5.4.65 RED_DX

- **Name:** Red (R) Value X Register
- **Description:** This register specifies the added Red (R) value for each step at X axis.
- **Base Address:** 0x40028000
- **Offset:** 0x1A0
- **Reset Value:** 0x00000000

Table 8-427 Red (R) Value X Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the added Red (R) value for each step at X axis.

8.5.5.4.66 RED_DY

- **Name:** Red (R) Value Y Register
- **Description:** This register specifies the added Red (R) value for each step at Y axis.
- **Base Address:** 0x40028000
- **Offset:** 0x1A4
- **Reset Value:** 0x00000000

Table 8-428 Red (R) Value Y Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the added Red (R) value for each step at Y axis.

8.5.5.4.67 GRE_DX

- **Name:** Green (G) Value X Register
- **Description:** This register specifies the added Green (G) value for each step at X axis.
- **Base Address:** 0x40028000
- **Offset:** 0x1A8
- **Reset Value:** 0x00000000

Table 8-429 Green (G) Value X Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the added Green (G) value for each step at X axis.

8.5.5.4.68 GRE_DY

- **Name:** Green (G) Value Y Register
- **Description:** This register specifies the added Green (G) value for each step at Y axis.
- **Base Address:** 0x40028000
- **Offset:** 0x1AC
- **Reset Value:** 0x00000000

Table 8-430 Green (G) Value Y Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the added Green (G) value for each step at Y axis.

8.5.5.4.69 BLU_DX

- **Name:** Blue (B) Value X Register
- **Description:** This register specifies the added Blue (B) value for each step at X axis.
- **Base Address:** 0x40028000
- **Offset:** 0x1B0
- **Reset Value:** 0x00000000

Table 8-431 Blue (B) Value X Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the added Blue (B) value for each step at X axis.

8.5.5.4.70 BLU_DY

- **Name:** Blue (B) Value Y Register
- **Description:** This register specifies the added Blue (B) value for each step at Y axis.
- **Base Address:** 0x40028000
- **Offset:** 0x1B4
- **Reset Value:** 0x00000000

Table 8-432 Blue (B) Value Y Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the added Blue (B) value for each step at Y axis.

8.5.5.4.71 ALF_DX

- **Name:** Alpha (A) Value X Register
- **Description:** This register specifies the added Alpha (A) value for each step at X axis.
- **Base Address:** 0x40028000
- **Offset:** 0x1B8
- **Reset Value:** 0x00000000

Table 8-433 Alpha (A) Value X Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the added Alpha (A) value for each step at X axis.

8.5.5.4.72 ALF_DY

- **Name:** Alpha (A) Value Y Register

- **Description:** This register specifies the added Alpha (A) value for each step at Y axis.
- **Base Address:** 0x40028000
- **Offset:** 0x1BC
- **Reset Value:** 0x00000000

Table 8-434 Alpha (A) Value Y Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the added Alpha (A) value for each step at Y axis.

8.5.5.4.73 RED_INIT

- **Name:** Red (R) Value Init Register
- **Description:** This register specifies the Red (R) value of the STARTXY pixel.
- **Base Address:** 0x40028000
- **Offset:** 0x1C0
- **Reset Value:** 0x00000000

Table 8-435 Red (R) Value Init Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the Red (R) value of the STARTXY pixel.

8.5.5.4.74 GRE_INIT

- **Name:** Green (G) Value Init Register
- **Description:** This register specifies the Green (G) value of the STARTXY pixel.
- **Base Address:** 0x40028000
- **Offset:** 0x1C4
- **Reset Value:** 0x00000000

Table 8-436 Green (G) Value Init Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the Green (G) value of the STARTXY pixel.

8.5.5.4.75 BLU_INIT

- **Name:** Blue (B) Value Init Register
- **Description:** This register specifies the Blue (B) value of the STARTXY pixel.

- **Base Address:** 0x40028000
- **Offset:** 0x1C8
- **Reset Value:** 0x00000000

Table 8-437 Blue (B) Value Init Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the Blue (B) value of the STARTXY pixel.

8.5.5.4.76 ALF_INIT

- **Name:** Alpha (A) Value Init Register
- **Description:** This register specifies the Alpha (A) value of the STARTXY pixel.
- **Base Address:** 0x40028000
- **Offset:** 0x1CC
- **Reset Value:** 0x00000000

Table 8-438 Alpha (A) Value Init Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Specifies the Alpha (A) value of the STARTXY pixel.

8.5.5.4.77 ROPBLENDER_BLEND_MODE

- **Name:** Blending Mode Register
- **Description:** This register specifies the blending modes for different calculations between the source (Fragment Processing Core) and destination (Framebuffer) colors.
- **Base Address:** 0x40028000
- **Offset:** 0x1D0
- **Reset Value:** 0x00000000

Table 8-439 Blending Mode Register

Bits	Field Name	RW	Reset	Description
31		RW	0x0	If set, destination color keying is enabled.
30		RW	0x0	If set, source color keying is enabled.
29:26	RSVD	R		Reserved bits
25		RW	0x0	If set, negative destination color keying is enabled.
24:12	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
11:8		RW	0x0	Indicates the blending mode of the stored pixel in the Framebuffer (destination pixel).
7:4	RSVD	R		Reserved bits
3:0		RW	0x0	Indicates the blending mode of the current pixel (source pixel).

8.5.5.4.78 ROPBLENDER_DST_CKEY

- **Name:** Destination Color Register
- **Description:** This register specifies the destination color value.
- **Base Address:** 0x40028000
- **Offset:** 0x1D4
- **Reset Value:** 0x00000000

Table 8-440 Destination Color Register

Bits	Field Name	RW	Reset	Description
31:24		RW	0x0	Specifies the Alpha (A) value.
23:16		RW	0x0	Specifies the Blue (B) value.
15:8		RW	0x0	Specifies the Green (G) value.
7:0		RW	0x0	Specifies the Red (R) value.

8.5.5.4.79 ROPBLENDER_CONST_COLOR

- **Name:** Constant Color Register
- **Description:** This register specifies the constant color value.
- **Base Address:** 0x40028000
- **Offset:** 0x1D8
- **Reset Value:** 0x00000000

Table 8-441 Constant Color Register

Bits	Field Name	RW	Reset	Description
31:24		RW	0x0	Specifies the Alpha (A) value.
23:16		RW	0x0	Specifies the Blue (B) value.
15:8		RW	0x0	Specifies the Green (G) value.
7:0		RW	0x0	Specifies the Red (R) value.

8.5.5.4.80 IP_VERSION

- **Name:** GPU Version Register
- **Description:** This register shows the current version of the GPU.
- **Base Address:** 0x40028000
- **Offset:** 0x1DC
- **Reset Value:** 0x00000000

Table 8-442 GPU Version Register

Bits	Field Name	RW	Reset	Description
31:0		R	0x0	Indicates the current version of the GPU.

8.5.5.4.81 IDREG

- **Name:** GPU ID Register
- **Description:** This register shows the GPU ID (fixed value).
- **Base Address:** 0x40028000
- **Offset:** 0x1EC
- **Reset Value:** 0x86362000

Table 8-443 GPU ID Register

Bits	Field Name	RW	Reset	Description
31:0		R	0x86362000	Indicates that the fixed value is 0x86362000.

8.5.5.4.82 CFG

- **Name:** GPU Configuration Register
- **Description:** This register shows the GPU configuration status.
- **Base Address:** 0x40028000
- **Offset:** 0x1F0
- **Reset Value:** 0x74030105

Table 8-444 GPU Configuration Register

Bits	Field Name	RW	Reset	Description
31		RW	0x0	Being set to 1 indicates that the master bus is AXI. Being set to 0 indicates that the master bus is AHB.
30		RW	0x1	Being set to 1 indicates that Bilinear filtering is enabled.

Bits	Field Name	RW	Reset	Description
29		RW	0x1	Being set to 1 indicates that TSC6 compression mode is enabled.
28		RW	0x1	Being set to 1 indicates that H/W Blender is enabled.
27		RW	0x0	When set to 1, bus to memory is asynchronous to core clock.
26		RW	0x1	When set to 1, dirty region is enabled.
25:18	RSVD	R		Reserved bits
17		RW	0x1	When set to 1, TSC4 compression mode is enabled.
16		RW	0x1	When set to 1, clock gating is enabled.
15:12	RSVD	R		Reserved bits
11:8		RW	0x1	Indicates the number of cores.
7:0		RW	0x5	Indicates the log2 number of threads.

8.5.5.4.83 CFG_H

- **Name:** GPU Configuration (High Bits) Register
- **Description:** This register shows the GPU configuration status (high bits).
- **Base Address:** 0x40028000
- **Offset:** 0x1F4
- **Reset Value:** 0x000000E3

Table 8-445 GPU Configuration (High Bits) Register

Bits	Field Name	RW	Reset	Description
31:8	RSVD	R		Reserved bits
7		RW	0x1	Indicates that external debug interface (GP_FLAG/ SYSERROR_IRQ/FREEZE) is enabled.
6		RW	0x1	Indicates that separate logic/mem clock gating is enabled.
5		RW	0x1	Indicates that texture map unit has two channels.
4		RW	0x0	Indicates that YUV coefficients are present.
3		RW	0x0	Indicates that Gamma Correction is enabled.
2		RW	0x0	Indicates that 10-bit precision is selected.
1		RW	0x1	Indicates that the TSC Framebuffer/Texture Decompression is enabled.
0		RW	0x1	Indicates that the antialiasing feature is enabled.

8.5.5.4.84 C0_REG

- **Name:** Reg0 RGBA Value Register
- **Description:** This register specifies the RGBA value stored in constant register 0 for fragment calculations.
- **Base Address:** 0x40028000
- **Offset:** 0x200
- **Reset Value:** 0x00000000

Table 8-446 Reg0 RGBA Value Register

Bits	Field Name	RW	Reset	Description
31:24		W	0x0	Specifies the Alpha (A) value.
23:16		W	0x0	Specifies the Blue (B) value.
15:8		W	0x0	Specifies the Green (G) value.
7:0		W	0x0	Specifies the Red (R) value.

8.5.5.4.85 C1_REG

- **Name:** Reg1 RGBA Value Register
- **Description:** This register specifies the RGBA value stored in constant register 1 for fragment calculations.
- **Base Address:** 0x40028000
- **Offset:** 0x204
- **Reset Value:** 0x00000000

Table 8-447 Reg1 RGBA Value Register

Bits	Field Name	RW	Reset	Description
31:24		W	0x0	Specifies the Alpha (A) value.
23:16		W	0x0	Specifies the Blue (B) value.
15:8		W	0x0	Specifies the Green (G) value.
7:0		W	0x0	Specifies the Red (R) value.

8.5.5.4.86 C2_REG

- **Name:** Reg2 RGBA Value Register
- **Description:** This register specifies the RGBA value stored in constant register 2 for fragment calculations.
- **Base Address:** 0x40028000
- **Offset:** 0x208
- **Reset Value:** 0x00000000

Table 8-448 Reg2 RGBA Value Register

Bits	Field Name	RW	Reset	Description
31:24		W	0x0	Specifies the Alpha (A) value.
23:16		W	0x0	Specifies the Blue (B) value.
15:8		W	0x0	Specifies the Green (G) value.
7:0		W	0x0	Specifies the Red (R) value.

8.5.5.4.87 C3_REG

- **Name:** Reg3 RGBA Value Register
- **Description:** This register specifies the RGBA value stored in constant register 3 for fragment calculations.
- **Base Address:** 0x40028000
- **Offset:** 0x20C
- **Reset Value:** 0x00000000

Table 8-449 Reg3 RGBA Value Register

Bits	Field Name	RW	Reset	Description
31:24		W	0x0	Specifies the Alpha (A) value.
23:16		W	0x0	Specifies the Blue (B) value.
15:8		W	0x0	Specifies the Green (G) value.
7:0		W	0x0	Specifies the Red (R) value.

8.5.5.4.88 COORD_MASK

- **Name:** Inverted Mask Register
- **Description:** This register specifies the inverted mask for rasterizer coordinates.
- **Base Address:** 0x40028000
- **Offset:** 0x4D0
- **Reset Value:** 0x00000FFF

Table 8-450 Inverted Mask Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0xFFF	Specifies the inverted mask for rasterizer coordinates.

8.5.5.4.89 IRQ_ID

- **Name:** Interrupt Signal Register

- **Description:** This register signals an interrupt.
- **Base Address:** 0x40028000
- **Offset:** 0xFF0
- **Reset Value:** 0x00000000

Table 8-451 Interrupt Signal Register

Bits	Field Name	RW	Reset	Description
31:0		RW	0x0	Writes any value to signal an interrupt.

8.5.5.4.90 GP_FLAGS

- **Name:** Stop/Break Register
- **Description:** This register stops/breaks the command list when debugging.
- **Base Address:** 0x40028000
- **Offset:** 0xFF4
- **Reset Value:** 0x00000000

Table 8-452 Stop/Break Register

Bits	Field Name	RW	Reset	Description
31:20	RSVD	R		Reserved bits
19		RW	0x0	When set to 1, set breakpoint for GP_FLAG line 0 if NEMA_GP_FLAGS[3] bit value is high.
18		RW	0x0	When set to 1, set breakpoint for GP_FLAG line 0 if NEMA_GP_FLAGS[2] bit value is high.
17		RW	0x0	When set to 1, set breakpoint for GP_FLAG line 0 if NEMA_GP_FLAGS[1] bit value is high.
16		RW	0x0	When set to 1, set breakpoint for GP_FLAG line 0 if NEMA_GP_FLAGS[0] bit value is high.
15:12	RSVD	R		Reserved bits
11		RW	0x0	GP_FLAG_irqmask when set to 1 masks/enables the generation of the IRQ_SYSERROR from line 3.
10		RW	0x0	GP_FLAG_irqmask when set to 1 masks/enables the generation of the IRQ_SYSERROR from line 2.
9		RW	0x0	GP_FLAG_irqmask when set to 1 masks/enables the generation of the IRQ_SYSERROR from line 1.

Bits	Field Name	RW	Reset	Description
8		RW	0x0	GP_FLAG_irqmask when set to 1 masks/enables the generation of the IRQ_SYSERROR from line 0.
7		RW	0x0	GP_FLAG_mask when set to 1 masks/enables the generation of the output FREEZE signal from line 3.
6		RW	0x0	GP_FLAG_mask when set to 1 masks/enables the generation of the output FREEZE signal from line 2.
5		RW	0x0	GP_FLAG_mask when set to 1 masks/enables the generation of the output FREEZE signal from line 1.
4		RW	0x0	GP_FLAG_mask when set to 1 masks/enables the generation of the output FREEZE signal from line 0.
3		RW	0x0	GP_FLAG line 3
2		RW	0x0	GP_FLAG line 2
1		RW	0x0	GP_FLAG line 1
0		RW	0x0	GP_FLAG line 0

8.5.5.4.91 SYS_INT

- **Name:** System Interrupt Register
- **Description:** This register indicates the type of the system interrupt.
- **Base Address:** 0x40028000
- **Offset:** 0xFF8
- **Reset Value:** 0x00000000

Table 8-453 System Interrupt Register

Bits	Field Name	RW	Reset	Description
31:12	RSVD	R		Reserved bits
11		RW	0x0	Indicates that a bus error has occurred.
10:7		RW	0x0	Indicates the bus error code interface. • 1000: AHB/AXI slave port • 0100: AHB/AXI M0 master port • 0010: AHB/AXI M1 master port • 0001: AXI CL master port
6:5		RW	0x0	Indicates the bus error code occurrence direction (AXI only). • 01: Read direction • 10: Write direction

Bits	Field Name	RW	Reset	Description
4		RW	0x0	Indicates the interface error code (AXI only). • 0: SLVERR • 1: DECERR
3		RW	0x0	Indicates that IRQ_SYSERROR occurs due to GP_FLAG line 3.
2		RW	0x0	Indicates that IRQ_SYSERROR occurs due to GP_FLAG line 2.
1		RW	0x0	Indicates that IRQ_SYSERROR occurs due to GP_FLAG line 1.
0		RW	0x0	Indicates that IRQ_SYSERROR occurs due to GP_FLAG line 0.

8.5.5.4.92 BUSERROR_MASK

- **Name:** Buserror Interrupt Mask Register
- **Description:** This register sets the mask for the Buserror Interrupt.
- **Base Address:** 0x40028000
- **Offset:** 0xFFC
- **Reset Value:** 0x00000000

Table 8-454 Buserror Interrupt Mask Register

Bits	Field Name	RW	Reset	Description
31:7	RSVD	R		Reserved bits
6:3		RW	0x0	Sets the mask for the bus error code interface. • 1000: AHB/AXI slave port • 0100: AHB/AXI M0 master port • 0010: AHB/AXI M1 master port • 0001: AXI CL master port
2:1		RW	0x0	Sets the mask for the bus error code occurrence direction (AXI only). • 01: Read direction • 10: Write direction
0	RSVD	R		Reserved bit

8.6 Audio Interface

8.6.1 PDM

8.6.1.1 Introduction

The Pulse Density Modulation (PDM) module provides 1 instance and 2 channels for external audio front ends, for example, digital microphones. It also supports DMA for data transfer.

8.6.1.2 Main Features

- 1 instance
- 2 channels
- Left channel: Collect data on the rising edge.
- Right channel: Collect data on the failing edge.
- Up to two PDM microphones configured as a left/right pair using the same data input
- Supporting 8 kHz, 15.625 kHz, and 16 kHz sampling rates
- 16-bit samples
- Hardware decimation filters
- DMA support
- Programmable gain control

8.6.1.3 Block Diagram

The functional block diagram of PDM is shown in Figure 8-103. The PDM module comprises DMIC Chain Left, DMIC Chain Right, and PDM Register Block.

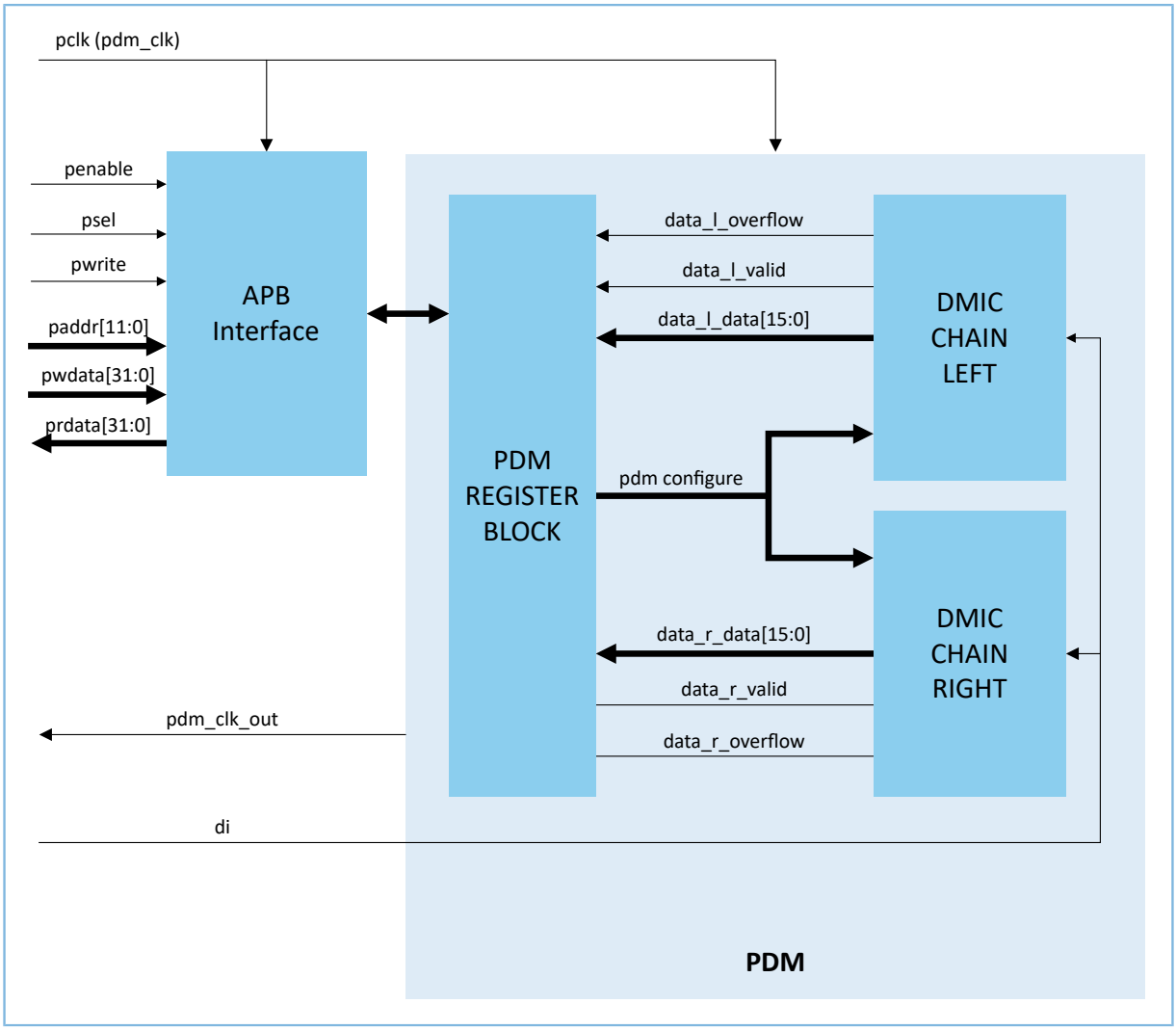

Figure 8-103 PDM block diagram

8.6.1.4 Protocols Introduction

PDM module is an interface for PDM signal input to PCM audio format output, so the input is 1-bit data in PDM format, and the output data is in PCM format.

8.6.1.4.1 Protocols Introduction

PDM is a modulation method that uses digital signals to represent analog signals. There is only 1-bit output, and non-zero is 1, so digital audio represented by PDM is also called oversampled 1-bit Audio. It is a high-frequency data stream of 1-bit digital sampling. In a PDM signal, the relative density of the pulse corresponds to the amplitude of the analog signal. As shown below, a large number of “1s” corresponds to high (positive) amplitude values, and a large number of “0s” corresponds to low (negative) amplitude values; alternating “1s” and “0s” corresponds to amplitude values.

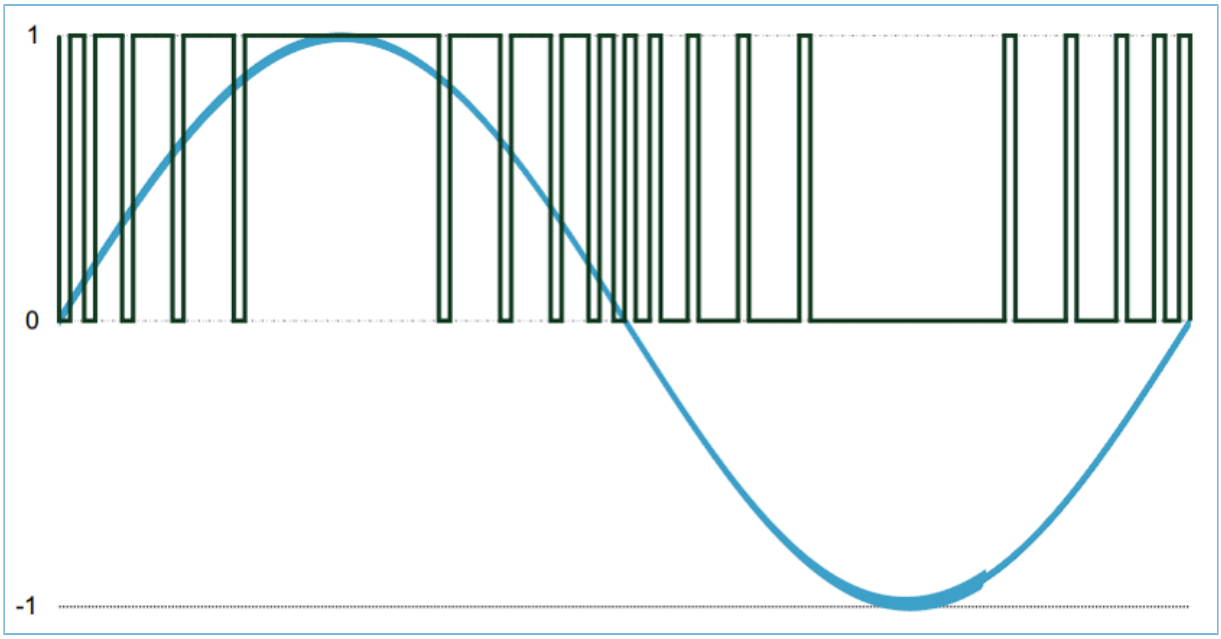

Figure 8-104 Pulse density modulation

8.6.1.4.2 PCM Introduction

In the PCM (Pulse Code Modulation) signal, the specific amplitude value is encoded as a pulse. The PCM stream has two basic properties which determine the fidelity of the stream relative to the original analog signal.

- Sampling rate
- Bit depth

Sampling rate is used to represent the signal digitally, the number of signal samples collected per second. The bit depth determines the number of bits of information in each sample.

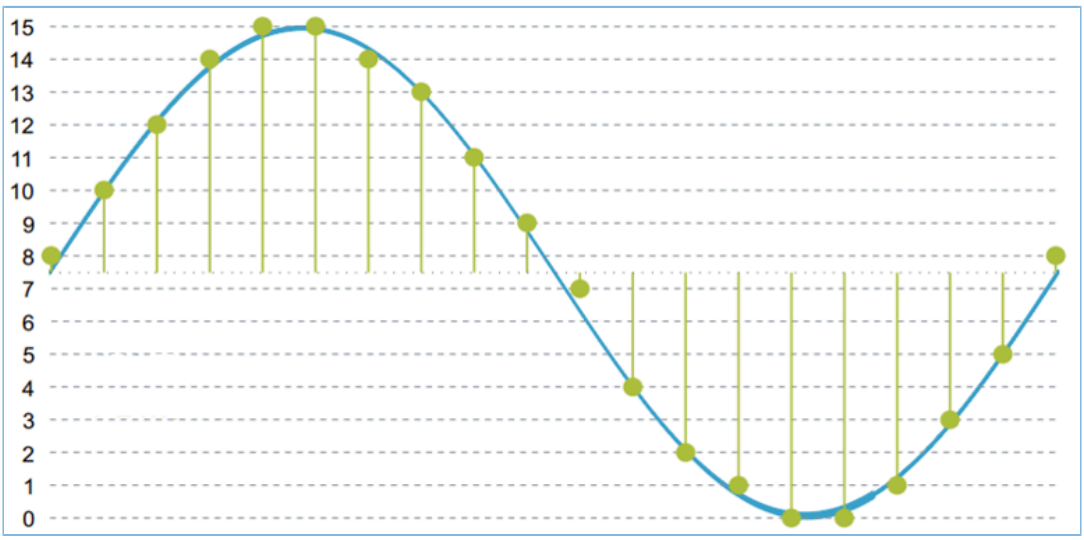

Figure 8-105 Pulse code modulation

8.6.1.5 Functional Description

The PDM module of GR5526 supports single-channel and dual-channel operating modes. It has two wires – serial data in (DI) and serial clock (CLOCK).

8.6.1.5.1 PDM Pin Mux

Before configuring the PDM module functions, make sure the pin mux of PDM interface GPIO is configured as PDM_CLKO and PDM_DI.

8.6.1.5.2 PDM Operation

By default, the left channel collects data on the rising edge of the clock, and the right channel collects data on the falling edge of the clock, resulting in two bit streams. In order to convert 1-bit data of PDM entrance into PCM audio samples, a decimation filter is included in the PDM module. The input of the filter is the two-channel PDM serial stream (with the left channel on clock high and the right channel on clock low). The output is 2 × 16-bit PCM samples at a sampling rate 64 times lower than the PDM clock rate.

8.6.1.5.3 PDM Gain

PDM module supports programmable gain control. The gain is controlled by the PGA_CFG_LEFT and PGA_CFG_RIGHT registers. PGA_CFG_LEFT controls the gain of the left channel, and PGA_CFG_RIGHT controls the gain of the right channel.

The control bit is a u4.10. This means it is a 14-bit unsigned number with 4 integer bits and 10 fractional bits. 0x400 gain is 0 dB, shift left by one bit approximately equal to +6dB; otherwise, shift right by one bit approximately equal to -6dB. For example, 0x800 = +6 dB and 0x200 = -6 dB. The conversion formula is as follows:

$$Gain = 20 * \log_{10}\left(\frac{X}{1024}\right) \quad (dB) \tag{28}$$

Where, X is the register value.

Some calculated values are given below:

Table 8-455 Calculated PDM gain values (examples)

Value (HEX)	Gain (dB)	Value (HEX)	Gain (dB)
00000000000001(0x01)	-60.2	00000100011111(0x11F)	-11.0
00000000000010(0x02)	-54.2	00000101000010(0x142)	-10.0
00000000000011(0x03)	-50.7	00000101101010(0x16A)	-9.0
00000000000100(0x04)	-48.2	00000110010110(0x196)	-8.0
00000000000101(0x05)	-46.2	00000111000111(0x1C7)	-7.0
00000000000110(0x06)	-44.6	00000111111111(0x1FF)	-6.0
00000000000111(0x07)	-43.3	00001000111101(0x23D)	-5.0
00000000001000(0x08)	-42.1	00001010000011(0x283)	-4.0
00000000001001(0x09)	-41.1	00001011010001(0x2D1)	-3.0
00000000001010(0x0A)	-40.2	00001100101001(0x329)	-2.0

Value (HEX)	Gain (dB)	Value (HEX)	Gain (dB)
00000000001011(0x0B)	-39.4	00001110001100(0x38C)	-1.0
00000000001100(0x0C)	-38.6	00010000000000(0x400)	0.0
00000000001110(0x0E)	-37.3	00010001110111(0x477)	1.0
00000000010000(0x10)	-36.1	00010100000010(0x502)	2.0
00000000010010(0x12)	-35.1	00010110011111(0x59F)	3.0
00000000010100(0x14)	-34.2	00011001001110(0x64E)	4.0
00000000010111(0x17)	-33.0	00011100010011(0x713)	5.0
00000000011001(0x19)	-32.2	00011111110000(0x7F0)	6.0
00000000011101(0x1D)	-31.0	00100011101000(0x8E8)	7.0
00000000100000(0x20)	-30.1	00100111111110(0x9FE)	8.0
00000000100100(0x24)	-29.1	00101010010101(0xA95)	8.5
00000000101001(0x29)	-28.0	00101100110110(0xB36)	9.0
00000000101110(0x2E)	-27.0	00110010010100(0xC94)	10.0
00000000110011(0x33)	-26.1	00111000011101(0xE1D)	11.0
00000000111001(0x39)	-25.1	00111111010110(0xFD6)	12.0
00000001000000(0x40)	-24.1	01000111000100(0x11C4)	13.0
00000001001000(0x48)	-23.1	01001111101111(0x13EF)	14.0
00000001010001(0x51)	-22.0	01011001011110(0x165E)	15.0
00000001011011(0x5B)	-21.0	01100100011000(0x1918)	16.0
00000001100110(0x66)	-20.0	01110000101000(0x1C28)	17.0
00000001110011(0x73)	-19.0	01111110011000(0x1F98)	18.0
00000010000001(0x81)	-18.0	10001101110011(0x2373)	19.0
00000010010000(0x90)	-17.0	10011111000110(0x27C6)	20.0
00000010100010(0xA2)	-16.0	10110010100000(0x2CA0)	21.0
00000010110110(0xB6)	-15.0	11001000010010(0x3212)	22.0
00000011001100(0xCC)	-14.0	11100000101110(0x382E)	23.0
00000011100100(0Xe4)	-13.0	11111100001001(0x3F09)	24.0
00000100000000(0x100)	-12.0		

8.6.1.5.4 PDM Enable

The PDM starts data sampling after the EN_LEFT/EN_RIGHT register and CLK registers have been written. When starting the module, it will take some time for the filters to start outputting valid data. Transients from the PDM microphone itself may also occur. The first few samples might hence contain invalid values or transients. It is therefore advised to discard the first few samples after a PDM start.

8.6.1.5.5 PDM Interrupt

The PDM module has two interrupts for the left and right channels respectively. One is the valid data interrupt, which will be generated when the PDM outputs data. The other is the data overflow interrupt, which will be generated when the DMA does not remove the data in time. The INT_LEFT/INT_RIGHT register can mask interrupts. The DATA_LEFT/DATA_RIGHT register can check interrupt flag.

8.6.1.5.6 PDM DMA

The PDM module supports one DMA channel for PDM transmit. It can carry data from three places: DATA_LEFT register, DATA_RIGHT register, and DATA register. The lower 16 bits of the DATA_LEFT register store the data of the left channel, and the lower 16 bits of the DATA_RIGHT register store the data of the right channel. The lower 16 bits of the DATA register store the left channel data, and the upper 16 bits store the right channel data. They are equivalent to the data of DATA_LEFT and DATA_RIGHT.

Note:

Since the PDM module does not have an internal FIFO, it is necessary to ensure the efficiency of the DMA as much as possible in the PDM DMA mode; otherwise the PDM is prone to overflow.

8.6.1.5.7 Hardware Connection Example

The PDM module is applicable to microphone voice acquisition. Connect the microphone clock to CLK, and data to DIN. The module supports three connection methods: left channel sample, right channel sample, and stereo sample, as shown below:

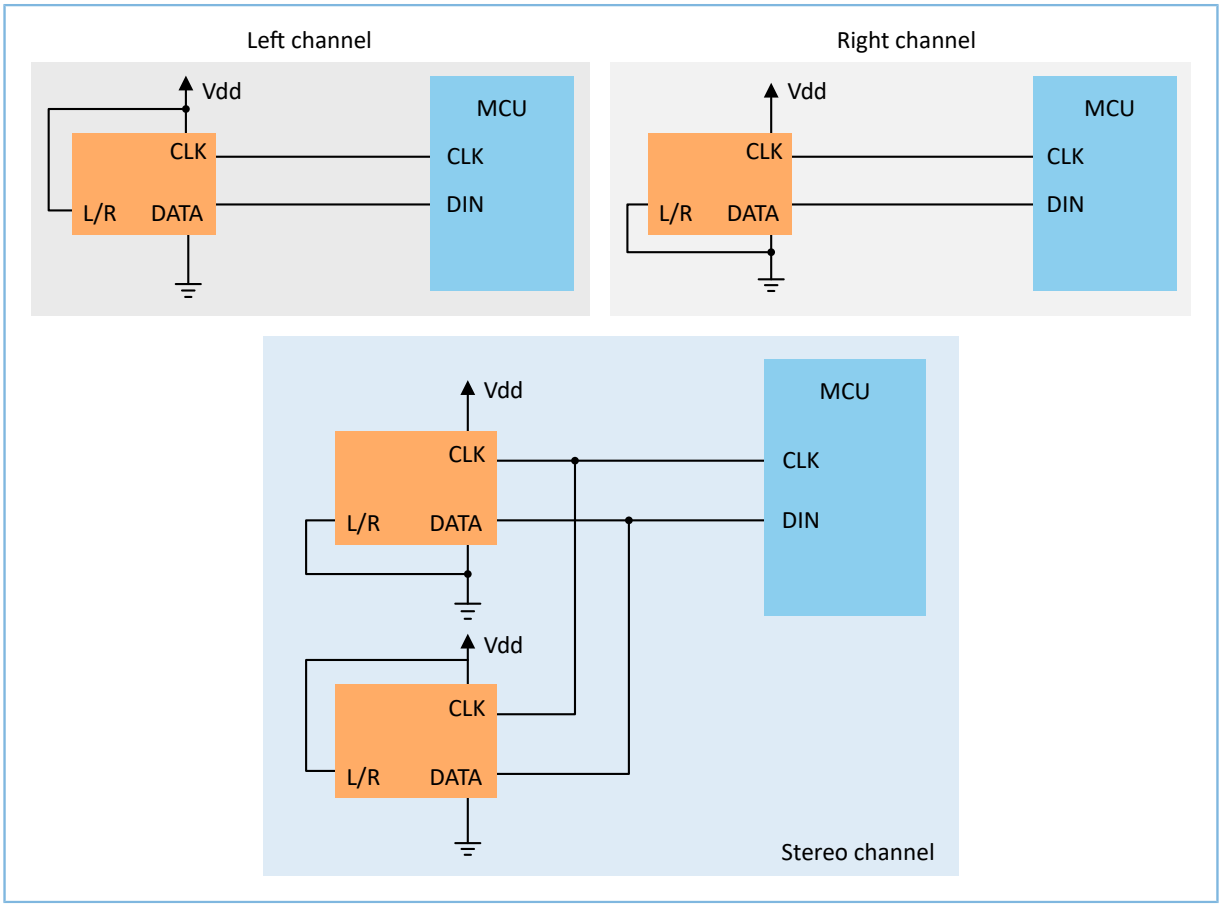

Figure 8-106 Hardware connection example

8.6.1.6 Registers

8.6.1.6.1 EN_LEFT

- **Name:** Left PDM Enable Register
- **Description:** This register specifies the sampling enable bit of the left channel.
- **Base Address:** 0x4000F700
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 8-456 Left PDM Enable Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	SAMPLE_DMIC	RW	0x0	Left channel sampling data • 0x0: Stop sampling data in. • 0x1: Start sampling data in.

Bits	Field Name	RW	Reset	Description
0	EN_RX	RW	0x0	Left channel reset • 0x0: Reset • 0x1: Not reset

8.6.1.6.2 PGA_CFG_LEFT

- **Name:** Left Channel Gain Register
- **Description:** This register specifies the gain of the left channel.
- **Base Address:** 0x4000F700
- **Offset:** 0x10
- **Reset Value:** 0x00000400

Table 8-457 Left Channel Gain Register

Bits	Field Name	RW	Reset	Description
31:14	RSVD	R		Reserved bits
0:13	VAL	RW	0x400	Left channel gain: It is a u4.10. This means it is a 14-bit unsigned number with 4 integer bits and 10 fractional bits. 0x400 gain is 0 dB, shift left by one bit (plus 3 dB); otherwise, shift right by one bit (minus 3 dB).

8.6.1.6.3 DATA_LEFT

- **Name:** Left Channel Data Register
- **Description:** This register specifies the data of the left channel.
- **Base Address:** 0x4000F700
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 8-458 Left Channel Data Register

Bits	Field Name	RW	Reset	Description
31:19	RSVD	R	0x0	Reserved bits
18	MASK	RW	0x0	Write 1 to mask dma_req.
17	OVER	RW	0x0	Left channel data overflow flag. If DMA does not read this register or combined data register before another sample is written, this bit gets asserted. Moreover, the bit can be used as an interrupt to MCU. Write a 1 to clear.

Bits	Field Name	RW	Reset	Description
16	VALID	R	0x0	Left channel data valid flag, also can be used as an interrupt to MCU. Read to clear.
15:0	DATA	R	0x0	Left channel data

8.6.1.6.4 INT_LEFT

- **Name:** Left Channel Interrupt Register
- **Description:** This register specifies the interrupt of the left channel.
- **Base Address:** 0x4000F700
- **Offset:** 0x18
- **Reset Value:** 0x00000001

Table 8-459 Left Channel Interrupt Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R	0x0	Reserved bits
1	OVER_MASK	RW	0x0	Left channel data overflow interrupt. Write 1 to mask the interrupt.
0	VALID_MASK	RW	0x1	Left channel data valid interrupt. Write 1 to mask the interrupt.

8.6.1.6.5 EN_RIGHT

- **Name:** Right PDM Enable Register
- **Description:** This register specifies the sampling enable bit of the right channel.
- **Base Address:** 0x4000F700
- **Offset:** 0x1C
- **Reset Value:** 0x00000000

Table 8-460 Right PDM Enable Register

Bits	Field Name	RW	Reset	Description
Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	SAMPLE_DMIC	RW	0x0	Right channel sampling data • 0x0: Stop sampling data in. • 0x1: Start sampling data in.
0	EN_RX	RW	0x0	Right channel reset

Bits	Field Name	RW	Reset	Description
				• 0x0: Reset • 0x1: Not reset

8.6.1.6.6 PGA_CFG_RIGHT

- **Name:** Right Channel Gain Register
- **Description:** This register specifies the gain of the right channel.
- **Base Address:** 0x4000F700
- **Offset:** 0x44
- **Reset Value:** 0x00000400

Table 8-461 Right Channel Gain Register

Bits	Field Name	RW	Reset	Description
31:14	RSVD	R		Reserved bits
0:13	VAL	RW	0x400	Right channel gain: It is a u4.10. This means it is a 14-bit unsigned number with 4 integer bits and 10 fractional bits. 0x400 gain is 0dB, shift left by one bit (plus 3 dB), otherwise, shift right by one bit (minus 3dB).

8.6.1.6.7 DATA_RIGHT

- **Name:** Right Channel Data Register
- **Description:** This register specifies the data of the right channel.
- **Base Address:** 0x4000F700
- **Offset:** 0x48
- **Reset Value:** 0x00000000

Table 8-462 Right Channel Data Register

Bits	Field Name	RW	Reset	Description
31:19	RSVD	R	0x0	Reserved bits
18	MASK	RW	0x0	Write 1 to mask dma_req.
17	OVER	RW	0x0	Right channel data overflow flag. If DMA does not read this register or combined data register before another sample is written, this bit gets asserted. Moreover, this bit can be used as an interrupt to MCU. Write a 1 to clear.

Bits	Field Name	RW	Reset	Description
16	VALID	R	0x0	Right channel data valid flag, also can be used as an interrupt to MCU. Read to clear.
0:15	DATA	R	0x0	Right channel data

8.6.1.6.8 INT_RIGHT

- **Name:** Right Channel Interrupt Register
- **Description:** This register specifies the interrupt of the right channel.
- **Base Address:** 0x4000F700
- **Offset:** 0x4C
- **Reset Value:** 0x00000001

Table 8-463 Right Channel Interrupt Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R	0x0	Reserved bits
1	OVER_MASK	RW	0x0	Right channel data overflow interrupt. Write 1 to mask the interrupt.
0	VALID_MASK	RW	0x1	Right channel data valid interrupt. Write 1 to mask the interrupt.

8.6.1.6.9 DATA

- **Name:** Data Register
- **Description:** This specifies the Data Register.
- **Base Address:** 0x4000F700
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 8-464 Data Register

Bits	Field Name	RW	Reset	Description
31:16	DATA_R	R	0x0	Right channel data, consistent with DATA_RIGHT register
15:0	DATA_L	R	0x0	Left channel data, consistent with DATA_LEFT register

8.6.1.6.10 CLK

- **Name:** CLK Enable Register
- **Description:** This specifies the CLK Enable Register.

- **Base Address:** 0x4000F700
- **Offset:** 0x58
- **Reset Value:** 0x00000000

Table 8-465 CLK Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R	0x0	Reserved bits
9:8	SAMPLE_RATE	RW	0x1	PDM sampling rate. • 0x0: 15.625K • 0x1: 16K • 0x3: 8K
0	EN	R	0x0	PDM CLK enable bit

8.6.1.7 Electrical Specifications

Table 8-466 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
$f_{Pdm,clk,64}$	PDM clock speed		1		M
$t_{PDM,JITTER}$	Jitter in PDM clock output			20	ns
$t_{PDM,DATA}$	Decimation filter delay			6.125	ms
$t_{PDM,cv}$	Allowed clock edge to data valid		56		ns
$t_{PDM,ci}$	Allowed (other) clock edge to data invalid		49		ns
$t_{PDM,s}$	Data setup time at fPDM,CLK = 1 MHz	65			ns
$t_{PDM,h}$	Data hold time at fPDM,CLK = 1 MHz	0			ns
$G_{PDM,default}$	Default (reset) absolute gain of the PDM module		0		dB

The PDM timing diagram is as below:

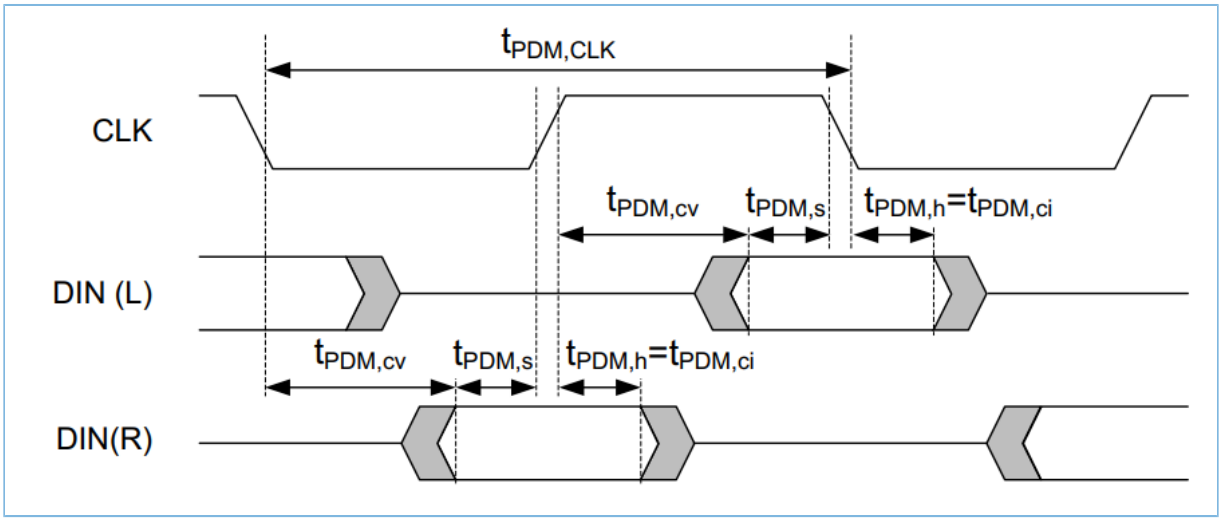

Figure 8-107 PDM timing diagram

8.6.2 I2S

8.6.2.1 Introduction

The I2S controller supports Inter-IC Sound Bus protocol to interface with external audio codec. Two 16-word deep FIFOs for read path and write path respectively are capable of handling 12-bit, 16-bit, 24-bit, and 32-bit word size. DMA controller handles the data movement between FIFO and memory.

A GR5526 series SoC has two I2S instances:

- I2S Master (I2S_M)
- I2S Slave (I2S_S)

I2S bus is used in systems that process digital audio signals, such as:

- A/D and D/A converters
- Digital signal processors
- Error correction for compact disc and digital recording
- Digital filters
- Digital I/O interfaces

I2S is a simple three-wire serial bus protocol developed by Philips to transfer stereo audio data. The bus only handles the transfer of audio data; hence, signals control and subcode need to be transferred separately using a different bus protocol (such as I2C).

8.6.2.2 Main Features

- 4-channel:
  - 1 RX channel and 1 TX channel for I2S Slave Controller

- 1 RX channel and 1 TX channel for I2S Master Controller

- Full duplex communication due to the independence of transmitter and receiver
- Asynchronous clocking of APB and I2S SCLK
- Audio data resolutions of 12 bits, 16 bits, 20 bits, 24 bits, and 32 bits
- SCLK gating for power saving
- Two 16-word FIFOs: one for transmission and the other for reception
- DMA support
- Programmable FIFO thresholds
- Various sampling rates: up to 96 kHz (16-bit WS)

8.6.2.3 Protocols Introduction

8.6.2.3.1 Signals and Formats

- The Inter-IC Sound (I2S) bus is a simple three-wire serial bus protocol developed by Philips to transfer stereo audio data. It is used extensively in audio interfaces. It adopts the design of data and clock signal separation, avoiding the distortion induced by time difference.
- The I2S bus has three signals:
  - SCLK: serial clock (also called BCLK. Each SCLK corresponds to each bit of audio data.)
  - WS: word select (also called LRCK. WS is used to switch between the left and right channel data.)
  - SD(SDATA): serial data (MSB → LSB: Data is transmitted from high-bit to low-bit)
- The I2S module of GR5526 adopts the I2S Philips standard (neither MSB justified standard nor LSB justified standard). For instance, when WS is low, the word being transferred is left stereo data; when WS is high, the word being transferred is right stereo data. For standard I2S formats, the MSB of a word is sent one SCLK cycle after a WS change. Serial data sent by the transmitter can be synchronized with either the negative edge or positive edge of the SCLK signal. However, the receiver must latch the serial data on the rising edge of SCLK. The formats (as receiver) are shown in Figure 8-108.

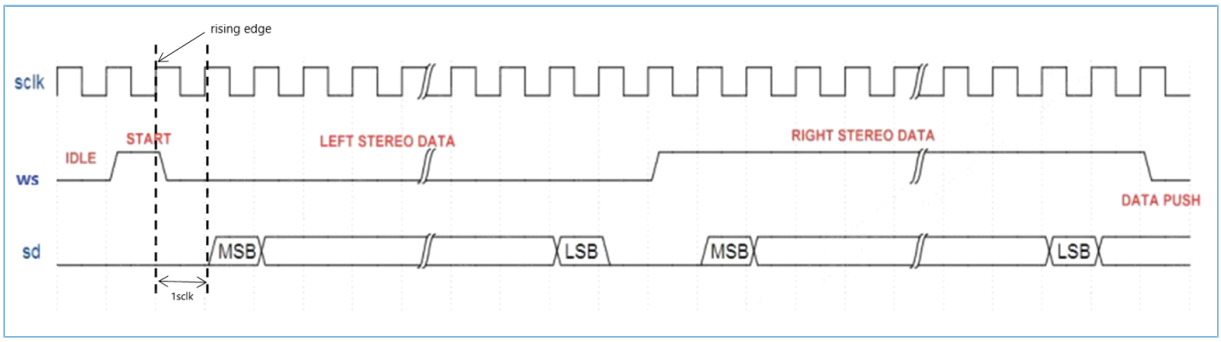

Figure 8-108 I2S formats (receiver)

8.6.2.3.2 Sampling and CLK frequency

As the frequency of SCLK is up to 3072 kHz, the number of SCLK cycles for which the (WS) line of I2S can be configured as 16 SCLK cycles, 24 SCLK cycles, or 32 SCLK cycles. If the 16 SCLK cycles are selected, the sampling rate is up to 96 kHz. The calculation formula is as follows:

- Sampling rate = Frequency of SCLK/[2 x Sampling depth (16,24 or 32)]
- Sampling rate = Frequency of WS

8.6.2.4 Functional Description

The I2S module of GR5526 supports full duplex communication thanks to the independence of transmitter and receiver. The module has four wires: serial data in (SDI), serial data out (SDO), word select (WS), and serial clock (SCLK).

I2S controller block diagram is illustrated in Figure 8-109.

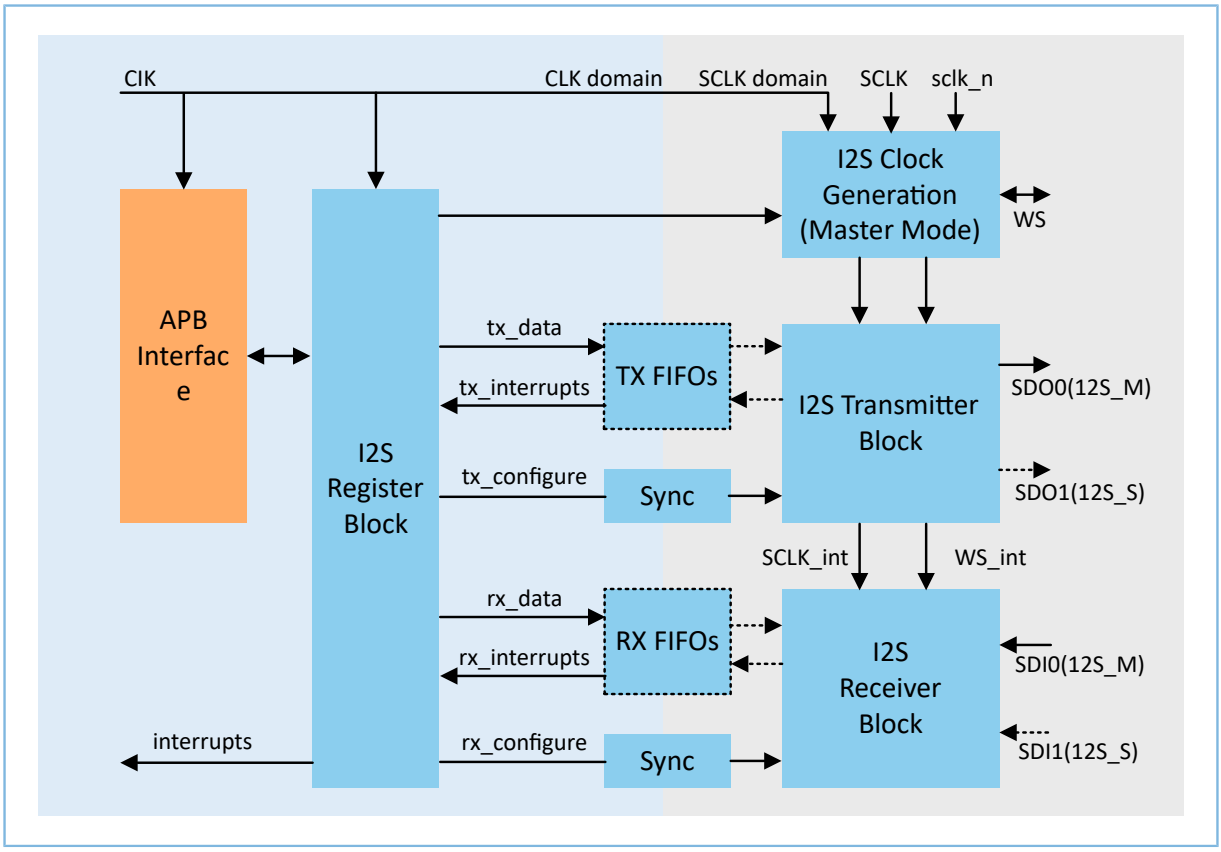

Figure 8-109 I2S block diagram

8.6.2.4.1 I2S Pin Mux

Before configuring the I2S module functions, the pin mux of four I2S interface GPIOs must be configured as I2S_M_SCLK (or I2S_S_SCLK), I2S_M_WS (or I2S_S_WS), I2S_M_SDI (or I2S_S_SDI), and I2S_M_SDO (or I2S_S_SDO).

8.6.2.4.2 Master and Slave

The I2S module can be used as a master or slave. The instance of master mode is I2S_M, and the instance of slave mode is I2S_S. When configured as a master, I2S controller initializes the WS signal and supplies the clock gating and clock enabling signals. Otherwise, when configured as a slave, I2S controller responds to externally generated SCLK and WS signals. When the I2S mode works in master mode, I2S clock generation must be enabled by setting the Clock Enable Register(CER) bit 0 to '1'.

8.6.2.4.3 Clock Divider

If the I2S module is used as master, the I2S Clock Configure Register (CLK_CFG) should be properly configured. The value of bit0~bit11 (div) is decided by the sampling rate, sampling depth (16, 24, or 32) and clock divider source (bit18). For example, if the CLK_CFG's bit18 is '1' (32 MHz) and the sampling rate is 48 kHz, the calculation formula is as follows:

```
div_TEMP0 = 32MHz *10/ (2 X Sampling Depth X Sample Rate)
if ((div_TEMP0 % 10) >=5)
{
div_TEMP1 = div_TEMP0 /10 +1;
}
else
{div_TEMP1 = div_TEMP0 /10;
}
if (div_TEMP1>= 2)
{
div = div_TEMP1: 2;
}
else
{
div = 0;
}
```

$$\text{Actual sampling rate} = \text{CLK_SRC} / \text{Actual divider} / \text{Bits} \tag{29}$$

Table 8-467 Sampling rate configuration

			CLK_SRC		**CLK_SRC**		**CLK_SRC**	
			32000K		**64000K**		**96000K**	
Sampling Rate (k)	**Bits**	**Desired Bit Clock (kHz)**	**Desired Divider (Value)**	**Actual Divider (Value)**	**Desired Divider (Value)**	**Actual Divider (Value)**	**Desired Divider (Value)**	**Actual Divider (Value)**
8	2*16 (left + right)	256	125	125	250	250	375	375
8	2*24 (left + right)	384	83.333333333	83	166.666666666	167	250	250
8	2*32 (left + right)	512	62.5	63	125	125	187.5	188
16	2*16 (left + right)	512	62.5	63	125	125	187.5	188

			CLK_SRC		CLK_SRC		CLK_SRC	
			32000K		64000K		96000K	
Sampling Rate (k)	Bits	Desired Bit Clock (kHz)	Desired Divider (Value)	Actual Divider (Value)	Desired Divider (Value)	Actual Divider (Value)	Desired Divider (Value)	Actual Divider (Value)
16	2*24 (left + right)	768	41.666666666	42	83.33333333	83	125	125
16	2*32 (left + right)	1024	31.25	31	62.5	63	93.75	94
48	2*16 (left + right)	1,536	20.83333333	21	41.666666667	42	62.5	63
48	2*24 (left + right)	2,304	13.88888889	14	27.777777778	28	41.666666667	42
48	2*32 (left + right)	3072	10.41666667	10	20.833333333	21	31.21951	31
44.1	2*16 (left + right)	1,411.2	22.67573696	23	45.351473923	45	68.02721088435	68
44.1	2*24 (left + right)	2,116.8	15.1171579743	15	30.234315969	30	45.3514739229	45
44.1	2*32 (left + right)	2,822.4	11.33786848	11	22.675736961	23	34.013605442	34

8.6.2.4.4 I2S Enable

You must enable the I2S controller before any data can be received or transmitted from or into the FIFOs. To enable the component, set the I2S enable (I2S_EN) bit of the I2S Enable Register (EN) to 1. When you disable the module, it acts as a global disable. To disable I2S controller, set EN[0] to 0.

After disable, the following events occur:

- TX and RX FIFOs are cleared, and read/write pointers are reset.
- Any data being transmitted or received is lost.
- All other programmable enables (such as transmitter/receiver block enables and individual TX/RX channel enables) in the component are overridden.

- Generation of master mode clock signals sclk_en, ws_out, and sclk_gate are disabled (for instance, they are held low).

When I2S controller is enabled and configured as a master, the device always starts in the left stereo data cycle (WS = 0), and one SCLK cycle after transitions to the right stereo data cycle (WS = 1). This allows for half a frame of SCLKS to write data to the TX FIFOs and to ensure that any connected slave receivers do not miss the start of the data frame (for instance, the WS 1: to-0 transition) once the SCLK restarts. (When I2S controller is configured as a slave, WS is externally supplied.) On reset, the EN[0] is set to 0 (disable).

8.6.2.4.5 I2S as Transmitter

I2S Master controller and I2S Slave controller support 1 I2S transmit (TX) channel. Stereo data pairs (such as, left and right audio data) written to a TX channel by CPU are shifted out serially on the appropriate serial data out line. The shifting is timed with respect to SCLK and WS.

Figure 8-110 illustrates the basic usage flow for an I2S controller when it acts as a transmitter.

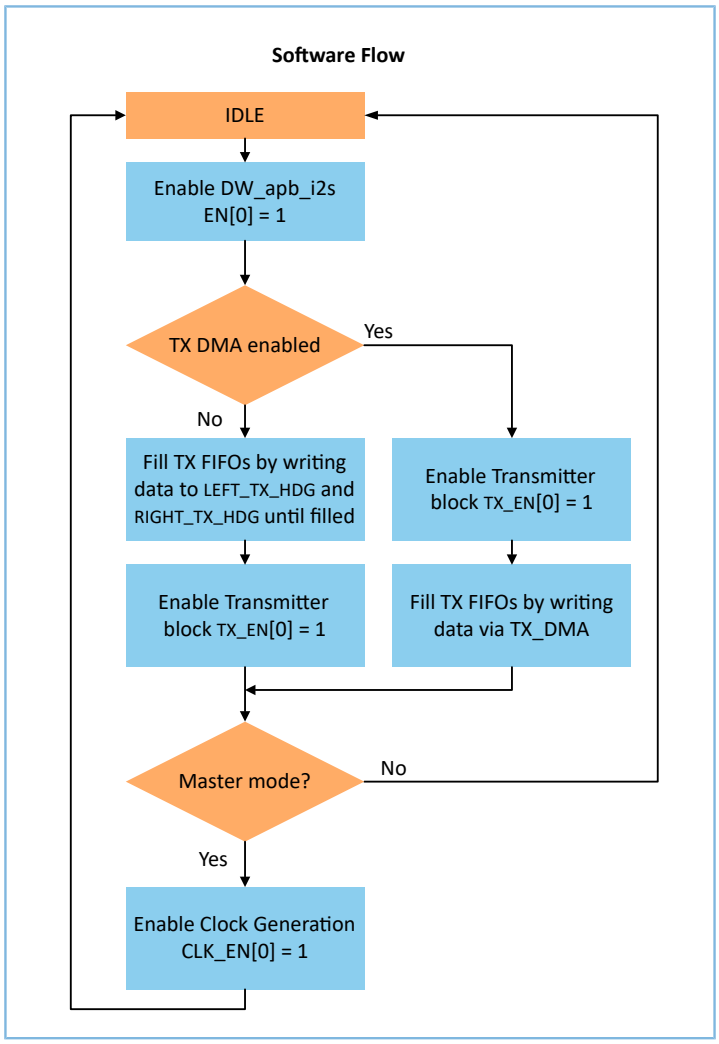

Figure 8-110 Basic usage flow when I2S as transmitter

8.6.2.4.6 I2S as Receiver

I2S Master controller and I2S Slave controller support 1 I2S receive (RX) channel. Stereo data pairs (such as, left and right audio data) are received serially from a data input line. These data words are stored in RX FIFOs until they are read by CPU. The receiving is timed with respect to SCLK and WS.

Figure 8-111 illustrates the basic usage flow for an I2S controller when it acts as a receiver.

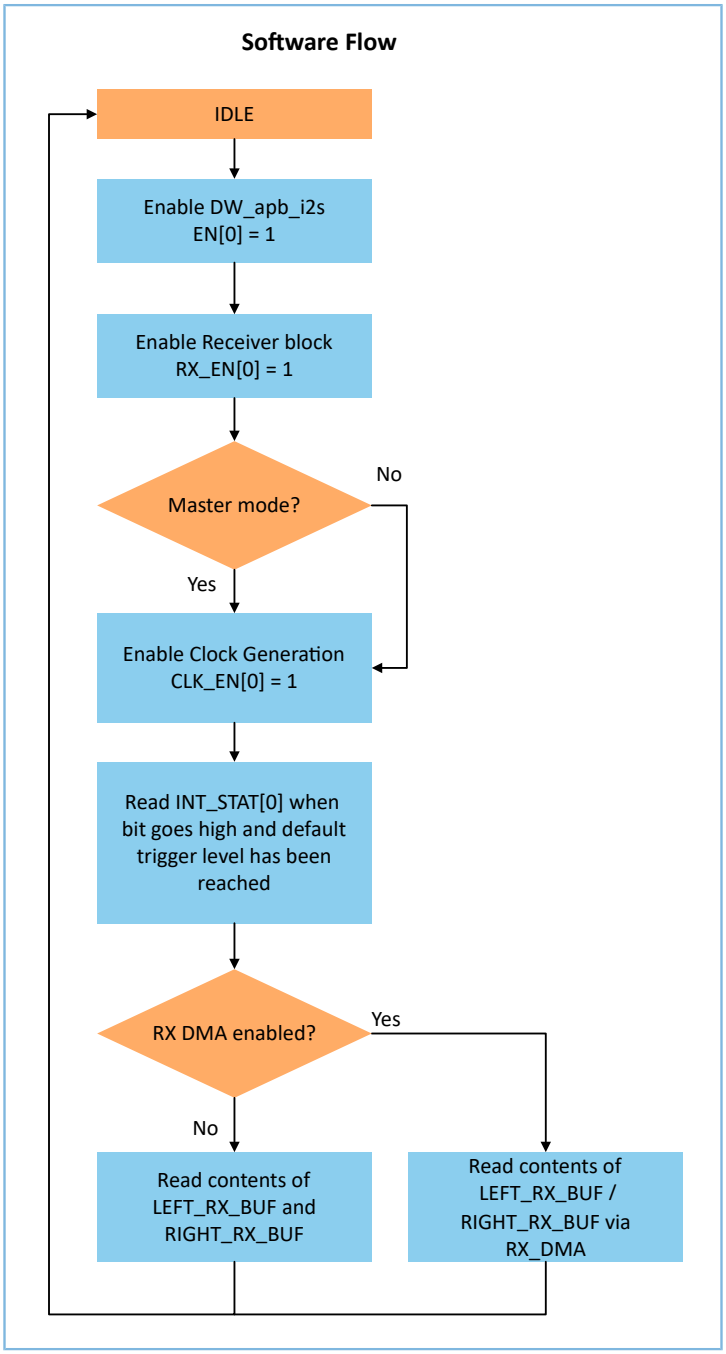

Figure 8-111 Basic usage flow when I2S as receiver

8.6.2.4.7 I2S FIFO

The I2S module provides two 16-word FIFOs, one for transmission and the other for reception. By configuring the Receive FIFO Configuration Register (RX_FIFO_CFG), you can program the trigger level in the RX FIFO at which the

Received Data Available Interrupt is generated. By configuring Transmit FIFO Configuration Register (TX_FIFO_CFG), you can program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt is generated. In addition, the Interrupt Mask Register (INT_MASK) is supported to mask TX/RX FIFO overrun and empty interrupts.

8.6.2.4.8 I2S DMA

The I2S module supports four DMA channels for I2S slave transmit, I2S slave receive, I2S master transmit, and I2S master receive respectively. The Receiver Block DMA Register (RX_DMA), Reset Receiver Block DMA Register (RST_RX_DMA), Transmitter Block DMA Register (TX_DMA), and Reset Transmitter Block DMA Register (RST_TX_DMA) are provided to configure these channels.

8.6.2.5 Registers

8.6.2.5.1 EN

- **Name:** I2S Enable Register
- **Description:** This register acts as a global enable/disable for I2S.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 8-468 I2S Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	I2S_EN	RW	0x0	I2S enable bit. This bit enables or disables I2S. A disable on this bit overrides any other block or channel as well as enables and flushes all FIFOs. **Value:** • 0x0: I2S disabled • 0x1: I2S enabled

8.6.2.5.2 RX_EN

- **Name:** I2S Receiver Block Enable Register
- **Description:** This register acts as an enable/disable for the I2S receiver block.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x4
- **Reset Value:** 0x00000000

Table 8-469 I2S Receiver Block Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RX_EN	RW	0x0	Receiver block enable. This bit enables or disables the receiver. A disable on this bit overrides any individual receive channel. **Value:** • 0x0: Receiver disabled • 0x1: Receiver enabled

8.6.2.5.3 TX_EN

- **Name:** I2S Transmitter Block Enable Register
- **Description:** This register acts as an enable/disable for the I2S transmitter block.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x8
- **Reset Value:** 0x00000000

Table 8-470 I2S Transmitter Block Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	TX_EN	RW	0x0	Transmitter block enable. This bit enables or disables the transmitter. A disable on this bit overrides any individual transmit channel. **Value:** • 0x0: Transmitter disabled • 0x1: Transmitter enabled

8.6.2.5.4 CLK_EN

- **Name:** Clock Enable Register
- **Description:** This register acts as an enable/disable for the I2S clock generation block.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0xC
- **Reset Value:** 0x00000000

Table 8-471 Clock Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	CLK_EN	RW	0x0	Clock generation enable/disable. This bit enables or disables the clock generation signals when I2S is a master. **Value:** • 0x0: Clock generation disabled • 0x1: Clock generation enabled **Note:** When the I2S is configured as a slave, this register serves no purpose.

8.6.2.5.5 SCLK_CFG

- **Name:** Clock Configuration Register
- **Description:** This register configures the WS when I2S is a master.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 8-472 Clock Configuration Register

Bits	Field Name	RW	Reset	Description
31:5	RSVD	R		Reserved bits
4:3	WS_SCLK	RW	0x0	These bits are used to program the number of SCLK cycles for which the word select line (ws_out) stays in the left or right sample mode. The I2S clock generation block must be disabled (CLK_EN [0] = 0) prior to any changes in this value. **Value:** • 0x0: 16 SCLK cycles • 0x1: 24 SCLK cycles • 0x2: 32 SCLK cycles
2:0	SCLK_GAT	RW	0x0	These bits are used to program the gating of SCLK. The programmed gating value must be less than or equal to the largest configured/programmed audio resolution to prevent the truncating of RX/TX data. The I2S clock generation

Bits	Field Name	RW	Reset	Description
				block must be disabled (CLK_EN[0] = 0) before making any changes in this value. **Values:** • 0x0 (NO_CLOCK_GATING): Clock gating is disabled • 0x1 (CLOCK_CYCLES_12): Gating after 12 SCLK cycles • 0x2 (CLOCK_CYCLES_16): Gating after 16 SCLK cycles • 0x3 (CLOCK_CYCLES_20): Gating after 20 SCLK cycles • 0x4 (CLOCK_CYCLES_24): Gating after 24 SCLK cycles Exists: This register is only relevant when component is configured to be a master (I2S_MODE_EN = 1).

8.6.2.5.6 RX_FIFO_RST

- **Name:** Receiver Block FIFO Reset Register
- **Description:** This register specifies the Receiver Block FIFO Reset Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 8-473 Receiver Block FIFO Reset Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	W		Reserved bits
0	RX_FIFO_RST	W	0x0	Receiver FIFO reset bit. Writing a 1 to this register flushes all the RX FIFOs (this is a self clearing bit). The receiver block must be disabled before writing to this bit. **Value:** • 0x0: Does not flush the RX FIFO. • 0x1: Flushes the RX FIFO.

8.6.2.5.7 TX_FIFO_RST

- **Name:** Transmitter Block FIFO Reset Register
- **Description:** This register specifies the Transmitter Block FIFO Reset Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)

- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 8-474 Transmitter Block FIFO Reset Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	W		Reserved bits
0	TX_FIFO_RST	W	0x0	Transmitter FIFO Reset. Writing a 1 to this register flushes all the TX FIFOs (this is a self clearing bit). The transmitter block must be disabled before writing to this bit. **Value:** • 0x0: Does not flush the TX FIFO. • 0x1: Flushes the TX FIFO.

8.6.2.5.8 LEFT_RX_BUF

- **Name:** Left Receive Buffer Register
- **Description:** This specifies the Left Receive Buffer Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 8-475 Left Receive Buffer Register

Bits	Field Name	RW	Reset	Description
31:0	LEFT_RX_BUF	R	0x0	The left stereo data received serially from the receive channel input (SDI) is read through this register. If the RX FIFO is full, and the two-stage read operation (for instance, a read from LEFT_RX_BUF followed by a read from RIGHT_RX_BUF) is not performed before the start of the next stereo pair, the new data is lost and an overrun interrupt occurs (data already in the RX FIFO is preserved). **Note:** Before reading this register again, the right stereo data must be read from RIGHT_RX_BUF; otherwise the status/interrupts will not be valid.

8.6.2.5.9 LEFT_TX_HDG

- **Name:** Left Transmit Holding Register
- **Description:** This specifies the Left Transmit Holding Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 8-476 Left Transmit Holding Register

Bits	Field Name	RW	Reset	Description
31:0	LEFT_TX_HDG	W	0x0	The left stereo data to be transmitted serially through the transmit channel output (SDO) is written through this register. Writing is a two-stage process: • A write to this register passes the left stereo sample to the transmitter. • A right stereo sample MUST be written to the RIGHT_TX_HDG register. Data must only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in data being lost and an overrun interrupt being generated.

8.6.2.5.10 RIGHT_RX_BUF

- **Name:** Right Receive Buffer Register
- **Description:** This specifies the Right Receive Buffer Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 8-477 Right Receive Buffer Register

Bits	Field Name	RW	Reset	Description
31:0	RIGHT_RX_BUF	R	0x0	The right stereo data received serially from the receive channel input (SDI) is read through this register. If the RX FIFO is full, and the two-stage read operation (for instance, read from LEFT_RX_BUF followed by a read from RIGHT_RX_BUF) is not performed before the start of the next stereo pair, the new data is lost and an overrun interrupt occurs (data already in the RX FIFO is preserved).

Bits	Field Name	RW	Reset	Description
				Note: Before reading this register, the left stereo data MUST be read from LEFT_RX_BUF; otherwise the status/interrupts will not be valid.

8.6.2.5.11 RIGHT_TX_HDG

- **Name:** Right Transmit Holding Register
- **Description:** This specifies the Right Transmit Holding Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 8-478 Right Transmit Holding Register

Bits	Field Name	RW	Reset	Description
31:0	RIGHT_TX_HDG	W	0x0	The right stereo data to be transmitted serially through the transmit channel output (SDO) is written through this register. Writing is a two-stage process: • A write to this register passes the right stereo sample to the transmitter. • A left stereo sample MUST be written to the LEFT_TX_HDG register. Data must only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in data being lost and an overrun interrupt being generated.

8.6.2.5.12 RX_CH_EN

- **Name:** Receive Enable Register
- **Description:** This specifies the Receive Enable Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x28
- **Reset Value:** 0x00000001

Table 8-479 Receive Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RX_CH_EN	RW	0x1	Receive channel enable bit. This bit enables/disables a receive channel. On enable, the channel begins receiving on the next left stereo cycle. A global disable of I2S (EN [0] = 0) or the receiver block (RX_EN [0] = 0) overrides this value. **Value:** • 0x0: Receive channel disabled • 0x1: Receive channel enabled

8.6.2.5.13 TX_CH_EN

- **Name:** Transmit Enable Register
- **Description:** This specifies the Transmit Enable Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x2c
- **Reset Value:** 0x00000001

Table 8-480 Transmit Enable Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	TX_CH_EN	RW	0x1	Transmit channel enable bit. This bit enables/disables a transmit channel. On enable, the channel begins transmitting on the next left stereo cycle. A global disable of I2S (EN [0] = 0) or transmitter block (TX_EN [0] = 0) overrides this value. **Value:** • 0x0: Transmit channel disabled • 0x1: Transmit channel enabled

8.6.2.5.14 RX_CFG

- **Name:** Receive Configuration Register

- **Description:** This specifies the Receive Configuration Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x30
- **Reset Value:** 0x00000005

Table 8-481 Receive Configuration Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	WORD_LEN	RW	0x5	These bits are used to program the desired data resolution of the receiver and enable the LSB of the incoming left (or right) word to be placed in the LSB of the LEFT_RX_BUF (or RIGHT_RX_BUF) register. Programmed data resolutions must be less than or equal to 0x5. If the selected resolution is greater than 0x5, the receive channel defaults go back to 0x5. The channel must be disabled prior to any changes in this value (RX_EN0[0] = 0). **Value:** • 0x0: Ignore the word length. • 0x1: 12-bit data resolution of the receiver • 0x2: 16-bit data resolution of the receiver • 0x3: 20-bit data resolution of the receiver • 0x4: 24-bit data resolution of the receiver • 0x5: 32-bit data resolution of the receiver

8.6.2.5.15 TX_CFG

- **Name:** Transmit Configuration Register
- **Description:** This specifies the Transmit Configuration Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x34
- **Reset Value:** 0x00000005

Table 8-482 Transmit Configuration Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits

Bits	Field Name	RW	Reset	Description
2:0	WORD_LEN	RW	0x5	These bits are used to program the data resolution of the transmitter and ensure the MSB of the data is transmitted first. Programmed resolutions must be less than or equal to 0x5. If the selected resolution is greater than 0x5, the transmit channel defaults go back to 0x5. The channel must be disabled prior to any changes in this value (TX_EN [0] = 0). **Value:** • 0x0: Ignore the word length. • 0x1: 12-bit data resolution of the transmitter • 0x2: 16-bit data resolution of the transmitter • 0x3: 20-bit data resolution of the transmitter • 0x4: 24-bit data resolution of the transmitter • 0x5: 32-bit data resolution of the transmitter

8.6.2.5.16 INT_STAT

- **Name:** Interrupt status Register
- **Description:** This specifies the Interrupt Status Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x38
- **Reset Value:** 0x00000010

Table 8-483 Interrupt status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	TX_FIFO_OVER	R	0x0	Status of data overrun interrupt for the TX channel. This bit specifies whether the TX FIFO write is valid or an overrun. Attempt to write to full TX FIFO. **Value:** • 0x0: TX FIFO write valid • 0x1: TX FIFO write overrun
4	TX_FIFO_EMPTY	R	0x1	Status of transmit empty trigger interrupt.

Bits	Field Name	RW	Reset	Description
				This bit specifies whether the TX FIFO trigger level has been reached or not. TX FIFO is empty. **Value:** • 0x0: TX FIFO trigger level is reached. • 0x1: TX FIFO trigger level is not reached.
3:2	RSVD	R		Reserved bits
1	RX_FIFO_OVER	R	0x0	Status of data overrun interrupt for the RX channel. Incoming data will be lost due to a full RX FIFO. **Value:** • 0x0: RX FIFO write valid • 0x1: RX FIFO write overrun
0	RX_DATA_AVL	R	0x0	Status of receive data available interrupt. This bit denotes the status of the RX FIFO trigger level. **Value:** • 0x1: RX FIFO trigger level is reached. • 0x0: RX FIFO trigger level is not reached.

8.6.2.5.17 INT_MASK

- **Name:** Interrupt Mask Register
- **Description:** This specifies the Interrupt Mask Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x3c
- **Reset Value:** 0x00000033

Table 8-484 Interrupt Mask Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	TX_FOM	RW	0x1	Masks TX FIFO overrun interrupts. This bit masks or unmasks a TX FIFO overrun interrupt. **Value:** • 0x1: Disable TX FIFO overrun interrupt. • 0x0: Enable TX FIFO overrun interrupt.
4	TX_FEM	RW	0x1	Masks TX FIFO empty interrupt. This bit masks or unmasks a TX FIFO empty interrupt.

Bits	Field Name	RW	Reset	Description
				Value: • 0x1: Disable TX FIFO Empty interrupt. • 0x0: Enable TX FIFO Empty interrupt.
3:2	RSVD	R		Reserved bits
1	RX_FOM	RW	0x1	Masks RX FIFO overrun interrupt. This bit masks or unmasks an RX FIFO overrun interrupt. **Value:** • 0x1: Disable RX FIFO overrun interrupt. • 0x0: Enable RX FIFO overrun interrupt.
0	RX_DAM	RW	0x1	Masks RX FIFO data available interrupt. This bit masks or unmasks an RX FIFO data available interrupt. **Value:** • 0x1: Disable RX FIFO data available interrupt. • 0x0: Enable RX FIFO data available interrupt.

8.6.2.5.18 RX_OVER

- **Name:** Receive Overrun Register
- **Description:** This specifies the Receive Overrun Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 8-485 Receive Overrun Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RX_CLR_FDO	R	0x0	Read this bit to clear the RX FIFO data overrun interrupt. **Value:** • 0x0: RX FIFO write valid • 0x1: RX FIFO write overrun

8.6.2.5.19 TX_OVER

- **Name:** Transmit Overrun Register
- **Description:** This specifies the Transmit Overrun Register.

- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x44
- **Reset Value:** 0x00000000

Table 8-486 Transmit Overrun Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	TX_CLR_FDO	R	0X0	Read this bit to clear the TX FIFO data overrun interrupt. **Value:** • 0x0: TX FIFO write valid • 0x1: TX FIFO write overrun

8.6.2.5.20 RX_FIFO_CFG

- **Name:** Receive FIFO Configuration Register
- **Description:** This specifies the Receive FIFO Configuration Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x48
- **Reset Value:** 0x00000008

Table 8-487 Receive FIFO Configuration Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	RX_FIFO_TL	RW	0x8	These bits program the trigger level in the RX FIFO at which the received data available interrupt is generated. Trigger Level = Programmed Value + 1 **Value:** • 0x0: Interrupt trigger when FIFO level is 1. • 0x1: Interrupt trigger when FIFO level is 2. • 0x2: Interrupt trigger when FIFO level is 3. • 0x3: Interrupt trigger when FIFO level is 4. • 0x4: Interrupt trigger when FIFO level is 5. • 0x5: Interrupt trigger when FIFO level is 6. • 0x6: Interrupt trigger when FIFO level is 7. • 0x7: Interrupt trigger when FIFO level is 8. • 0x8: Interrupt trigger when FIFO level is 9.

Bits	Field Name	RW	Reset	Description
				• 0x9: Interrupt trigger when FIFO level is 10. • 0xA: Interrupt trigger when FIFO level is 11. • 0xB: Interrupt trigger when FIFO level is 12. • 0xC: Interrupt trigger when FIFO level is 13. • 0xD: Interrupt trigger when FIFO level is 14. • 0xE: Interrupt trigger when FIFO level is 15. • 0xF: Interrupt trigger when FIFO level is 16.

8.6.2.5.21 TX_FIFO_CFG

- **Name:** Transmit FIFO Configuration Register
- **Description:** This specifies the Transmit FIFO Configuration Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x4c
- **Reset Value:** 0x00000008

Table 8-488 Transmit FIFO Configuration Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3:0	TX_FIFO_TL	RW	0x8	These bits program the trigger level in the TX FIFO at which the empty threshold reached interrupt is generated. Trigger Level = TXCHET **Value:** • 0x0: Interrupt trigger when FIFO level is 1. • 0x1: Interrupt trigger when FIFO level is 2. • 0x2: Interrupt trigger when FIFO level is 3. • 0x3: Interrupt trigger when FIFO level is 4. • 0x4: Interrupt trigger when FIFO level is 5. • 0x5: Interrupt trigger when FIFO level is 6. • 0x6: Interrupt trigger when FIFO level is 7. • 0x7: Interrupt trigger when FIFO level is 8. • 0x8: Interrupt trigger when FIFO level is 9. • 0x9: Interrupt trigger when FIFO level is 10. • 0xA: Interrupt trigger when FIFO level is 11. • 0xB: Interrupt trigger when FIFO level is 12.

Bits	Field Name	RW	Reset	Description
				• 0xC: Interrupt trigger when FIFO level is 13. • 0xD: Interrupt trigger when FIFO level is 14. • 0xE: Interrupt trigger when FIFO level is 15. • 0xF: Interrupt trigger when FIFO level is 16.

8.6.2.5.22 RX_FIFO_FLUSH

- **Name:** Receive FIFO Flush Register
- **Description:** This specifies the Receive FIFO Flush Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 8-489 Receive FIFO Flush Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	W		Reserved bits
0	RX_FIFO_RST	W	0x0	Receive channel FIFO reset. Writing a 1 to this register flushes an individual RX FIFO (this is a self clearing bit). An RX channel or block must be disabled prior to writing to this bit. **Value:** • 0x0: Does not flush an individual RX FIFO. • 0x1: Flushes an individual RX FIFO.

8.6.2.5.23 TX_FIFO_FLUSH

- **Name:** Transmit FIFO Flush Register
- **Description:** This specifies the Transmit FIFO Flush Register.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x54
- **Reset Value:** 0x00000000

Table 8-490 Transmit FIFO Flush Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	W		Reserved bits
0	TX_FIFO_RST	W	0x0	Transmit channel FIFO reset.

Bits	Field Name	RW	Reset	Description
				Writing a 1 to this register flushes an individual TX FIFO (this is a self clearing bit). A TX channel or block must be disabled prior to writing to this bit. **Value:** • 0x0: Does not flush an individual TX FIFO. • 0x1: Flushes an individual TX FIFO.

8.6.2.5.24 RX_DMA

- **Name:** Receiver Block DMA Register
- **Description:** The RX_DMA register allows access to receive channel via a single point rather than through the LEFT_RX_BUF and RIGHT_RX_BUF registers. The receive channel is targeted in a cyclical fashion (starting at the lowest numbered enabled channel) and takes two reads (left and right stereo data) before the component points to the next channel.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x1c0
- **Reset Value:** 0x00000000

Table 8-491 Receiver Block DMA Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	RX_DMA	RW	0x0	Receiver Block DMA Register. This bit is used to cycle repeatedly through the enabled receive channel, reading stereo data pairs.

8.6.2.5.25 RST_RX_DMA

- **Name:** Reset Receiver Block DMA Register
- **Description:** The RX_DMA can be reset to the enabled channel via the RST_RX_DMA register. The RST_RX_DMA register can be written to at any stage of the RX_DMA's read cycle; however, it has no effect when the component is in the middle of a stereo pair read.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x1c4
- **Reset Value:** 0x00000000

Table 8-492 Reset Receiver Block DMA Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	W		Reserved bits
0	RST_RX_DMA	W	0x0	Reset Receiver Block DMA Register. Writing a 1 to this self-clearing register resets the RX_DMA register mid-cycle to point to the enabled receive channel. **Value:** • 0x0: No effect. • 0x1: Reset receiver block DMA register.

8.6.2.5.26 TX_DMA

- **Name:** Transmitter Block DMA Register
- **Description**: The TX_DMA register functions in a similar way to the RX_DMA register and allows write accesses to all of the enabled transmit channels via a single point rather than through the LEFT_TX_HDG and RIGHT_TX_HDG registers.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x1c8
- **Reset Value:** 0x00000000

Table 8-493 Transmitter Block DMA Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	TX_DMA	RW	0x0	Transmitter Block DMA Register. This bit is used to cycle repeatedly through the enabled transmit channel to allow writing of stereo data pairs.

8.6.2.5.27 RST_TX_DMA

- **Name:** Reset Transmitter Block DMA Register
- **Description:** This register provides the same functionality as the **RST_RX_DMA** register but targets **TX_DMA** instead.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x1cc
- **Reset Value:** 0x00000000

Table 8-494 Reset Transmitter Block DMA Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	W		Reserved bits
0	RST_TX_DMA	W	0x0	Reset Transmitter Block DMA Register. Writing a 1 to this self-clearing register resets the TX_DMA register mid-cycle to point to the enabled transmit channel. **Value:** • 0x0: No effect. • 0x1: Reset transmitter block DMA register.

8.6.2.5.28 CLK_CFG

- **Name:** I2S Clock Configure Register
- **Description:** This register is used to configure I2S master clock.
- **Base Address:** 0x4000F200 (master) and 0x4000F000 (slave)
- **Offset:** 0x268
- **Reset Value:** 0x00040008

Table 8-495 I2S Clock Configure Register

Bits	Field Name	RW	Reset	Description
31:19	RSVD	R		Reserved bits
18	CLK_SRC_SEL	RW	0x1	Selects a clock divider source. • 0: 96 M • 1: 64 M • 2: 32 M
17	RSVD	R		Reserved bit
16	DIV_EN	RW	0x0	Enables I2S clock divider.
15:12	RSVD	R		Reserved bits
11:0	DIV	RW	0x8	Ratio = 1/(div + 2). Duty cycle is not 50 when the number is odd. For example: div = 0, ratio = 1/2

8.6.2.6 Electrical Specifications

Table 8-496 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
f_{SCK}	I2S clock frequency			3072	kHz
t_{SUSDI}	Data input setup time (from SDI to SCK)	20			ns
t_{HDSDI}	Data input hold time (from SCK to SDI change)	15			ns
t_{VSDO}	Data output valid time (from SCK to SDO valid)			20	ns
DC_{SCK}	SCK clock duty cycle	45		55	%

The I2S timing diagram is as below:

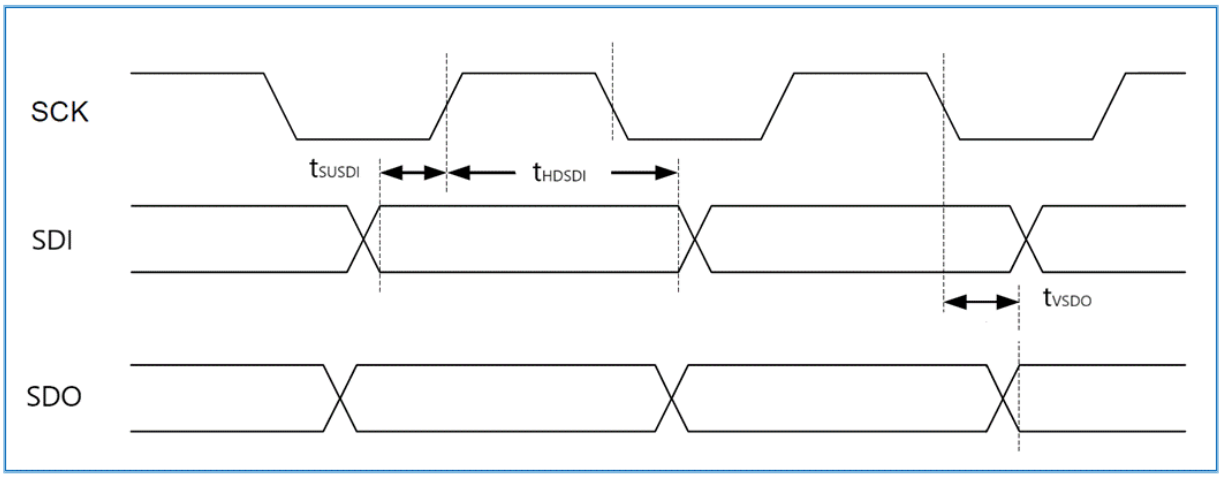

Figure 8-112 I2S timing diagram

8.7 Security Interfaces

8.7.1 KEYRAM

KEYRAM is mainly used for key derivation after power up. It has a memory area for storing key information. There is an embedded encryption and decryption hardware sub-module in KEYRAM to protect the key information.

In security mode, random numbers are generated as masks during each boot stage in order to prevent the external exhaustive method from cracking the key information. The random number masks are stored in KEYRAM.

When the security module needs to obtain the key from the KEYRAM through the key port bus, the KEYRAM embedded sub-module will automatically decrypt and import the key with the confusion of the random number mask into the security module. In addition, the security module also needs to load the random number mask so as to recovery the plain key.

8.7.2 Advanced Encryption Standard (AES)

8.7.2.1 Introduction

Advanced Encryption Standard (AES), also known as Rijndael encryption method in cryptography, is a block encryption standard adopted by the U.S. federal government. This standard is used to replace the Data Encryption Standard (DES). The block length of AES is fixed at 128 bits, and the key length can be 128, 192, or 256 bits. The basic processes of AES include SubBytes, ShiftRows, MixColumns, and AddRoundKey.

8.7.2.2 Main Features

- Compliance with NIST “Advanced Encryption Standard (AES), FIPS Publication 197” from November 2001
- Supports 128-bit, 192-bit, and 256-bit key.
- Supports encryption mode and decryption mode.
- Supports interrupt and query register to report status.
- Supports Electronic Codebook (ECB) and Cipher Block chaining (CBC) modes.
- Supports MCU and DMA operation.

8.7.2.3 Functional Description

8.7.2.3.1 Standard

Table 8-497 AES module standard

Document Code	Name of Standard
FIPS PUB 197	Advanced Encryption Standard (AES)

8.7.2.3.2 Module Overview

The AES co-processor encrypts or decrypts data, using an algorithm and implementation fully compliant with the AES defined in Federal Information Processing Standards (FIPS) publication 197. GR5526 embeds the AES module in order to realize hardware acceleration of calculation.

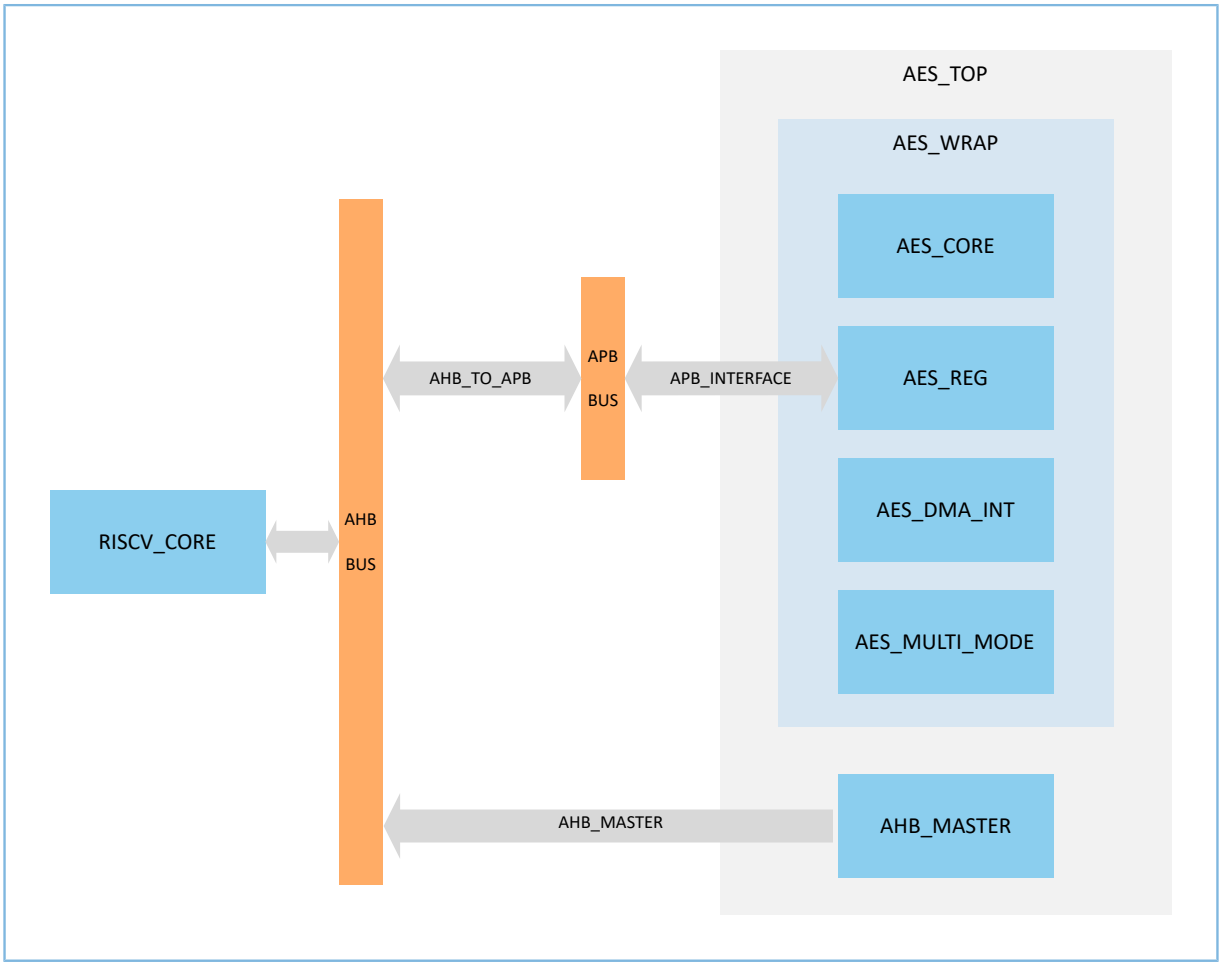

Figure 8-113 AES module in the system

Electronic Codebook (ECB) and Cipher Block Chaining (CBC) are supported for key size of 128 bits, 192 bits, and 256 bits.

Due to the feature of block cipher of AES, the data to be encrypted requires an integer multiple of 16 bytes.

The AES co-processor adopts both 32-bit AHB and APB interfaces. It supports DMA transfers for incoming and outgoing data (through a dedicated integrated DMA).

8.7.2.3.3 Usage

The CFG register can be used to configure relevant parameters for the AES module. CFG.KEY_MODE can be used to configure the key size to 128 bits, 192 bits, or 256 bits. CFG.DEC_ENC_SEL can be used to configure the encryption or decryption mode. CFG.OPT_MODE can be used to configure the working mode of ECB or CBC. CFG.KEY_TYPE can be used to configure the mode of fetching the key.

If the key is configured by MCU, the KEY0 to KEY7 registers need to be configured (128-bit for KEY0 to KEY3 only and 192-bit for KEY0 to KEY5 only). If the key is fetched through AHB, the KEY_ADDR register should be configured.

CTRL.FKEY_EN is configured in order to enable the AES module to fetch the key through AHB master interface. The STAT register is used to indicate the current status of the AES module. The random seeds are set to the INIT_SSI, INIT_SSO, MASK_SSI, and MASK_SSO registers.

The initialization vector (IV) should be set to the INIT_V0 to INIT_V3 registers for AES CBC mode.

If the AES module is in interrupt or DMA operation mode, the INT register needs to be configured.

In MCU operation mode, a block data (128-bit) is written into the DATA_IN0 to DATA_IN3 registers. In DMA operation mode, the XFER_SIZE, RD_START_ADDR, and WR_START_ADDR registers should be configured. The AES module will read the data from the address which is stored in the RD_START_ADDR register.

The AES module can be started through the CTRL register. The STAT register or the interrupt callback function can judge whether the calculation completes.

The plain data or the encrypted data can be read from the DATA_OUT0 to DATA_OUT3 registers in MCU operation mode. In DMA operation mode, the result has been sent to the address which is stored in the WR_START_ADDR register.

The endian of the data needs to be noticed and configured; otherwise it will affect the result of encryption and decryption. If the data is encrypted and decrypted in batches in CBC mode, the encryption result should be set as the IV to the INIT_V0 to INIT_V3 registers.

8.7.2.4 Registers

8.7.2.4.1 CTRL

- **Name:** AES Controller Register
- **Description:** This register acts as a global enable/disable for AES.
- **Base Address:** 0x40015000
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 8-498 AES Controller Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3	FKEY_EN	W	0x0	Enable AES fetch key by itself through AHB master interface or key port. This register can be cleared by itself when key_valid is set to 1.
2	DMA_MODE_EN	RW	0x0	DMA mode start enable bit. Start N block data encryption or decryption in DMA mode.
1	MCU_MODE_EN	RW	0x0	MCU mode start enable bit. Start a block data encryption or decryption in DMA mode. This signal should be cleared to zero before a new block input data is ready to start.
0	MODULE_EN	RW	0x0	Enables AES module. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0: Disable • 0x1: Enable

8.7.2.4.2 CFG

- **Name:** AES Configuration Register
- **Description:** This register acts as a configuration for AES.
- **Base Address:** 0x40015000
- **Offset:** 0x4
- **Reset Value:** 0x00000000

Table 8-499 AES Configuration Register

Bits	Field Name	RW	Reset	Description
31:13	RSVD	R		Reserved bits
12:11	KEY_TYPE	RW	0x0	Key type selection for encryption/decryption **Value:** • 0x0: Configured by MCU (default) • 0x1: Fetched through AHB interface • 0x2: Fetched through key port • 0x3: Reserved
10:8	OPT_MODE	RW	0x0	Selection for operation mode **Value:** • 0x0: Electronic Codebook (ECB) mode • 0x1: Cipher Block Chaining (CBC) mode • 0x2–0x7: Reserved for future application
7	ENDIAN	RW	0x0	Selection for data endian control **Value:** • 0x0: Reverse to big-endian (default) • 0x1: No reverse
6	FIRST_BLK	W	0x0	This register should be set to 1 before starting the first block in normal CBC and DMA CBC modes, and this register will be cleared by itself.
5	LOAD_SEED	W	0x0	Loads seed for LFSR. Seed for LFSR will be reloaded when this register is written with 1. This register can be cleared by itself.

Bits	Field Name	RW	Reset	Description
4	DEC_ENC_SEL	RW	0x0	Selection for encryption/decryption **Value:** • 0x0: Decryption (default) • 0x1: Encryption
3	FULL_MASK_EN	RW	0x0	Full mask enable signal **Value:** • 0x0: Disable (default) • 0x1: Enable (DPA resistance)
2	RSVD	R		Reserved bit
1:0	KEY_MODE	RW	0x0	Key mode selection for encryption/decryption **Value:** • 0x0: 128 bits (default) • 0x1: 192 bits • 0x2: 256 bits • 0x3: Reserved

8.7.2.4.3 STAT

- **Name:** AES Status Register
- **Description:** This is a read-only register used to indicate the current status of AES.
- **Base Address:** 0x40015000
- **Offset:** 0x8
- **Reset Value:** 0x00000000

Table 8-500 AES Status Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3	KEY_STAT	R	0x0	Key status, whether the key is ready to read or not **Value:** • 0x0: Not ready • 0x1: Ready
2	DMA_XFE_ERR	R WC	0x0	DMA transfer error. Write 1 to clear this register. **Value:** • 0x0: No error • 0x1: Error

Bits	Field Name	RW	Reset	Description
1	DMA_XFE_CPLT	R	0x0	DMA transfer complete **Value:** • 0x0: Not complete • 0x1: Complete
0	READY	R	0x0	AES result data out ready or not. This signal may be cleared when MCUEN (CTRL[1]) is disabled. **Value:** • 0x0: Not ready • 0x1: Ready

8.7.2.4.4 INT

- **Name:** AES Interrupt Register
- **Description:** This register enables or disables all interrupts generated by the AES. Additionally, it reports the status of the AES interrupts after they have been enabled.
- **Base Address:** 0x40015000
- **Offset:** 0x0C
- **Reset Value:** 0x00000000

Table 8-501 AES Interrupt Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	CPLT_INT_EN	RW	0x0	AES result data complete interrupt **Value:** • 0x0: Disable • 0x1: Enable
0	CPLT_INT_FLAG	R WC	0x0	AES result data complete interrupt flag. Write 1 to clear. **Read:** • 0x0: Not interrupt • 0x1: Interrupt **Write:** • 0x0: Not effect • 0x1: Clear

8.7.2.4.5 XFE_SIZE

- **Name:** AES Transfer Size Register
- **Description:** This register sets a size for AES DMA mode transfer.
- **Base Address:** 0x40015000
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 8-502 AES Transfer Size Register

Bits	Field Name	RW	Reset	Description
31:15	RSVD	R		Reserved bits
14:0	SIZE	RW	0x0	Total transfer size. Up to 32 KB • 0x000f: 1 block • 0x001f: 2 blocks • 0x002f: 3 blocks … • 0x7fff: 2048 blocks

8.7.2.4.6 RD_START_ADDR

- **Name:** AES Read Start Address Register
- **Description:** This register acts as a start address for AES DMA reading.
- **Base Address:** 0x40015000
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 8-503 AES Read Start Address Register

Bits	Field Name	RW	Reset	Description
31:0	ADDR	RW	0x0	In DMA mode, sets the read start address of transfer.

8.7.2.4.7 WR_START_ADDR

- **Name:** AES Write Start Address Register
- **Description:** This register acts as a start address for AES DMA writing.
- **Base Address:** 0x40015000
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 8-504 AES Write Start Address Register

Bits	Field Name	RW	Reset	Description
31:0	ADDR	RW	0x0	In DMA mode, sets the write start address of transfer.

8.7.2.4.8 KEY_ADDR

- **Name:** AES Key Address Register
- **Description:** This register acts as a key address for AES encryption or decryption.
- **Base Address:** 0x40015000
- **Offset:** 0x1C
- **Reset Value:** 0x00000000

Table 8-505 AES Key Address Register

Bits	Field Name	RW	Reset	Description
31:0	ADDR	RW	0x0	Indicates the AES key address in memory.

8.7.2.4.9 DATA_OUT0

- **Name:** AES Data Output 0 Register
- **Description:** This register is a 32-bit read-only buffer for the result data[127:96] from AES.
- **Base Address:** 0x40015000
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 8-506 AES Data Output 0 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_OUT0	R	0x0	AES result data[127:96]

8.7.2.4.10 DATA_OUT1

- **Name:** AES Data Output 1 Register
- **Description:** This register is a 32-bit read-only buffer for the result data[95:64] from AES.
- **Base Address:** 0x40015000
- **Offset:** 0x24
- **Reset Value:** 0x00000000

Table 8-507 AES Data Output 1 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_OUT1	R	0x0	AES result data[95:64]

8.7.2.4.11 DATA_OUT2

- **Name:** AES Data Output 2 Register
- **Description:** This register is a 32-bit read-only buffer for the result data[63:32] from AES.
- **Base Address:** 0x40015000
- **Offset:** 0x28
- **Reset Value:** 0x00000000

Table 8-508 AES Data Output 2 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_OUT2	R	0x0	AES result data[63:32]

8.7.2.4.12 DATA_OUT3

- **Name:** AES Data Output 3 Register
- **Description:** This register is a 32-bit read-only buffer for the result data[31:0] from AES.
- **Base Address:** 0x40015000
- **Offset:** 0x2C
- **Reset Value:** 0x00000000

Table 8-509 AES Data Output 3 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_OUT3	R	0x0	AES result data[31:0]

8.7.2.4.13 KEY0

- **Name:** AES Key 0 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[255:224] for AES.
- **Base Address:** 0x40015000
- **Offset:** 0x30
- **Reset Value:** 0x00000000

Table 8-510 AES Key 0 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY0	W	0x0	AES key[255:224]

8.7.2.4.14 KEY1

- **Name:** AES Key 1 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[223:192] for AES.
- **Base Address:** 0x40015000
- **Offset:** 0x34
- **Reset Value:** 0x00000000

Table 8-511 AES Key 1 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY1	W	0x0	AES key[223:192]

8.7.2.4.15 KEY2

- **Name:** AES Key 2 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[191:160] for AES.
- **Base Address:** 0x40015000
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 8-512 AES Key 2 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY2	W	0x0	AES key[191:160]

8.7.2.4.16 KEY3

- **Name:** AES Key 3 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[159:128] for AES.
- **Base Address:** 0x40015000
- **Offset:** 0x3C
- **Reset Value:** 0x00000000

Table 8-513 AES Key 3 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY3	W	0x0	AES key[159:128]

8.7.2.4.17 KEY4

- **Name:** AES Key 4 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[127:96] for AES.
- **Base Address:** 0x40015000
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 8-514 AES Key 4 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY4	W	0x0	AES key[127:96]

8.7.2.4.18 KEY5

- **Name:** AES Key 5 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[95:64] for AES.
- **Base Address:** 0x40015000
- **Offset:** 0x44
- **Reset Value:** 0x00000000

Table 8-515 AES Key 5 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY5	W	0x0	AES key[95:64]

8.7.2.4.19 KEY6

- **Name:** AES Key 6 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[63:32] for AES.
- **Base Address:** 0x40015000
- **Offset:** 0x48
- **Reset Value:** 0x00000000

Table 8-516 AES Key 6 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY6	W	0x0	AES key[63:32]

8.7.2.4.20 KEY7

- **Name:** AES Key 7 Register
- **Description:** This register is a 32-bit write-only buffer for the key data[31:0] for AES.
- **Base Address:** 0x40015000
- **Offset:** 0x4C
- **Reset Value:** 0x00000000

Table 8-517 AES Key 7 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY7	W	0x0	AES key[31:0]

8.7.2.4.21 INIT_SSI

- **Name:** AES Sbox Initial Seed Input Register
- **Description:** This register acts as an initial seed input for Sbox.
- **Base Address:** 0x40015000
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 8-518 AES Sbox Initial Seed Input Register

Bits	Field Name	RW	Reset	Description
31:0	SEED	RW	0x0	Sbox initial seed input

8.7.2.4.22 INIT_SSO

- **Name:** AES Sbox Initial Seed Output Register
- **Description:** This register acts as an initial seed output for Sbox.
- **Base Address:** 0x40015000
- **Offset:** 0x54
- **Reset Value:** 0x00000000

Table 8-519 AES Sbox Initial Seed Output Register

Bits	Field Name	RW	Reset	Description
31:0	SEED	RW	0x0	Sbox initial seed output

8.7.2.4.23 MASK_SSI

- **Name:** AES Sbox Seed Input Mask Register
- **Description:** This register acts as a mask for Sbox initial seed input mask.
- **Base Address:** 0x40015000
- **Offset:** 0x58
- **Reset Value:** 0x00000000

Table 8-520 AES Sbox Seed Input Mask Register

Bits	Field Name	RW	Reset	Description
31:0	MASK	RW	0x0	Sbox initial seed input mask

8.7.2.4.24 MASK_SSO

- **Name:** AES Sbox Seed Output Mask Register
- **Description:** This register acts as a mask for Sbox initial seed output mask.
- **Base Address:** 0x40015000
- **Offset:** 0x5C
- **Reset Value:** 0x00000000

Table 8-521 AES Sbox Seed Output Mask Register

Bits	Field Name	RW	Reset	Description
31:0	MASK	RW	0x0	Sbox initial seed output mask

8.7.2.4.25 INIT_V0

- **Name:** AES Initialization Vector 0 Register
- **Description:** This register acts as an initialization vector[127:96] for CBC mode.
- **Base Address:** 0x40015000
- **Offset:** 0x60
- **Reset Value:** 0x00000000

Table 8-522 AES Initialization Vector 0 Register

Bits	Field Name	RW	Reset	Description
31:0	VECTOR	W	0x0	Initialization vector[127:96] for CBC mode

8.7.2.4.26 INIT_V1

- **Name:** AES Initialization Vector 1 Register
- **Description:** This register acts as an initialization vector[95:64] for CBC mode.
- **Base Address:** 0x40015000
- **Offset:** 0x64
- **Reset Value:** 0x00000000

Table 8-523 AES Initialization Vector 1 Register

Bits	Field Name	RW	Reset	Description
31:0	VECTOR	W	0x0	Initialization vector[95:64] for CBC mode

8.7.2.4.27 INIT_V2

- **Name:** AES Initialization Vector 2 Register
- **Description:** This register acts as an initialization vector[63:32] for CBC mode.
- **Base Address:** 0x40015000
- **Offset:** 0x68
- **Reset Value:** 0x00000000

Table 8-524 AES Initialization Vector 2 Register

Bits	Field Name	RW	Reset	Description
31:0	VECTOR	W	0x0	Initialization vector[63:32] for CBC mode

8.7.2.4.28 INIT_V3

- **Name:** AES Initialization Vector 3 Register
- **Description:** This register acts as an initialization vector[31:0] for CBC mode.
- **Base Address:** 0x40015000
- **Offset:** 0x6C
- **Reset Value:** 0x00000000

Table 8-525 AES Initialization Vector 3 Register

Bits	Field Name	RW	Reset	Description
31:0	VECTOR	W	0x0	Initialization vector[31:0] for CBC mode

8.7.2.4.29 DATA_IN0

- **Name:** AES Data Input 0 Register
- **Description:** This register is a 32-bit only write buffer for the input data[127:96] for encryption or decryption.
- **Base Address:** 0x40015000
- **Offset**: 0x70
- **Reset Value:** 0x00000000

Table 8-526 AES Data Input 0 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_IN0	W	0x0	Input data[127:96] for encryption or decryption. This register is valid in MCU mode.

8.7.2.4.30 DATA_IN1

- **Name:** AES Data Input 1 Register
- **Description:** This register is a 32-bit only write buffer for the input data[95:64] for encryption or decryption.
- **Base Address:** 0x40015000
- **Offset:** 0x74
- **Reset Value:** 0x00000000

Table 8-527 AES Data Input 1 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_IN1	W	0x0	Input data[95:64] for encryption or decryption. This register is valid in MCU mode.

8.7.2.4.31 DATA_IN2

- **Name:** AES Data Input 2 Register
- **Description:** This register is a 32-bit only write buffer for the input data[63:32] for encryption or decryption.
- **Base Address:** 0x40015000
- **Offset:** 0x78
- **Reset Value:** 0x00000000

Table 8-528 AES Data Input 2 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_IN2	W	0x0	Input data[63:32] for encryption or decryption. This register is valid in MCU mode.

8.7.2.4.32 DATA_IN3

- **Name:** AES Data Input 3 Register
- **Description:** This register is a 32-bit only write buffer for the input data[31:0] for encryption or decryption.
- **Base Address:** 0x40015000
- **Offset:** 0x7C
- **Reset Value:** 0x00000000

Table 8-529 AES Data Input 3 Register

Bits	Field Name	RW	Reset	Description
31:0	DATA_IN3	W	0x0	Input data[31:0] for encryption or decryption. This register is valid in MCU mode.

8.7.2.4.33 KEYPORT_MASK

- **Name:** AES Keyport Mask Register
- **Description:** This register acts as a mask for key from key port.
- **Base Address:** 0x40015000
- **Offset:** 0x80
- **Reset Value:** 0x00000000

Table 8-530 AES Keyport Mask Register

Bits	Field Name	RW	Reset	Description
31:0	MASK	W	0x0	Mask for key from key port

8.7.3 Hash-based Message Authentication Code (HMAC)

8.7.3.1 Introduction

Hash-based Message Authentication Code (HMAC) is a mechanism for message authentication by using hash functions in cryptography. The message authentication provided by HMAC includes two aspects:

- Message integrity authentication: This can prove that the message has not been modified during transmission.

- Source identity authentication: Because both sides of the communication share the authentication key, receiver can verify that the source sending the data is consistent with the claim so that the receiver can reliably confirm that the received message is consistent with the sent message.

8.7.3.2 Main Features

- Compatible with SHA-256
- Supports custom initial hash values.
- Supports querying registers and interrupts to report status.
- Supports MCU mode and DMA mode.

8.7.3.3 Functional Description

8.7.3.3.1 Standard

Table 8-531 HMAC module standard

Document Code	Name of Standard
FIPS PUB 180-4	Secure Hash Standard (SHS) (Compliant with SHA-256)
FIPS PUB 198-1	Keyed-Hash Message Authentication Code (HMAC)

8.7.3.3.2 Module Overview

The HMAC co-processor authenticates data, using an algorithm and implementation fully compliant with the Keyed-Hash Message Authentication Code defined in Federal Information Processing Standards (FIPS) Publication 198. GR5526 embeds the HMAC module in order to realize hardware acceleration of calculation.

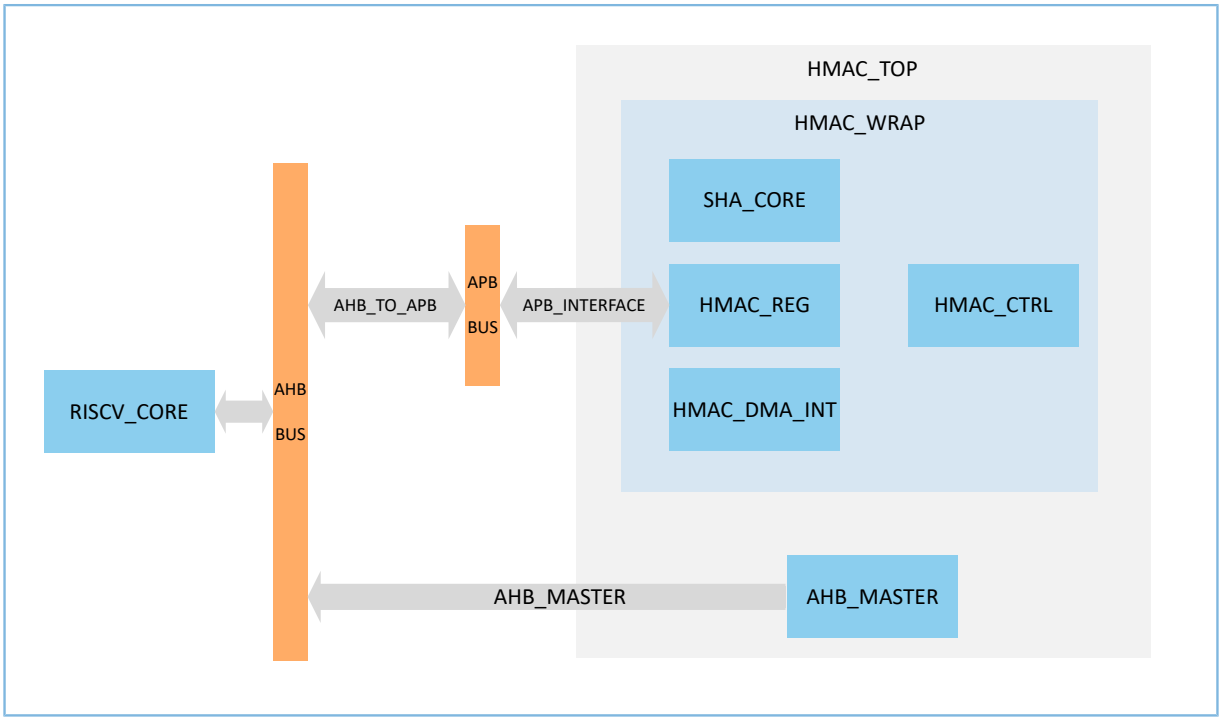

Figure 8-114 HMAC module in the system

Multiple modes are supported (SHA256, HMAC-SHA256) for a key size of 256 bits.

Information preprocessing is required for input data, including additional padding bits and additional length value.

The HMAC co-processor adopts both 32-bit AHB and APB interfaces. It supports DMA transfers for incoming and outgoing data (through a dedicated integrated DMA).

8.7.3.3.3 Usage

The CFG register can be used to configure relevant parameters for the HMAC module. CFG.CALC_TYPE can be used to configure the working mode including SHA-256 and HMAC. If the working mode is set to HMAC, the mode of fetching the key should be configured through CFG.KEY_TYPE.

If the key is configured by MCU, the KEY0 to KEY7 registers need to be configured. If the key is fetched through AHB, the KEY_ADDR register should be configured.

The SER_HASH0 to USER_HASH7 registers are used to customize the initial HASH values if the CFG register enables users to define the initial HASH.

If the HMAC module is in interrupt or DMA operation mode, the INT register needs to be configured.

In DMA operation mode, the XFER_SIZE, RD_START_ADDR, and WR_START_ADDR registers should be configured. If misalignment is required when MCU reads or writes data, the last two bits of the RD_START_ADDR and the WR_START_ADDR registers can be configured.

The HMAC module can be started through the CTRL register. CTRL.KEY_EN is configured in order to enable the HMAC module to fetch the key through AHB master interface.

In MCU operation mode, a block data (512-bit) is continuously written into the DATA_IN register. In DMA operation mode, the HMAC module will read the data from the address which is stored in the RD_START_ADDR register.

The STAT register or interrupt callback function can judge whether the calculation completes. Before writing next block data into the DATA_IN register, the INT register needs to be cleared.

In MCU operation mode, the result of the SHA-256 or HMAC can be obtained by repeatedly reading the DATA_OUT register until getting 256-bit data. In DMA operation mode, the result has been sent to the address which is stored in the WR_START_ADDR register.

8.7.3.4 Registers

8.7.3.4.1 CTRL

- **Name:** HMAC Controller Register
- **Description:** This register acts as a global enable/disable for HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 8-532 HMAC Controller Register

Bits	Field Name	RW	Reset	Description
31:4	RSVD	R		Reserved bits
3	LST_TX	W	0x0	Last block in MCU mode or last DMA transfer. This signal may be cleared by itself when hmac_ready was set.
2	KEY_EN	W	0x0	Enable HMAC fetch key by itself through AHB master interface or key port. This signal may be cleared by itself when hmac_key_valid was set.
1	DMA_START	RW	0x0	DMA mode start enable. Starting DMA transfer, this signal should be cleared after DMA has finished all transfers.
0	EN	RW	0x0	HMAC enable, high valid for whole HMAC processing. Users must disable it once all HMAC blocks complete. This signal should be asserted until all blocks complete.

8.7.3.4.2 CFG

- **Name:** HMAC Configuration Register
- **Description:** This register acts as a configuration for HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x4
- **Reset Value:** 0x00000000

Table 8-533 HMAC Configuration Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	PRIVT_MOD	RW	0x0	To resist DPA, select private mode. **Value:** • 0x0: Standard mode • 0x1: Private mode
4	CALC_TYPE	RW	0x0	Selects a calculation type. **Value:** • 0x0: HMAC • 0x1: SHA
3:2	KEY_TYPE	RW	0x0	Selects a key type. **Value:** • 0x0: Configured by MCU • 0x1: Fetched through AHB interface • 0x2: Fetched through key port • 0x3: Reserved
1	ENDIAN	RW	0x0	Selection for endian control **Value:** • 0x0: Reverse to big-endian (default) • 0x1: No reverse
0	HASH	RW	0x0	Enables user-defined hash. **Value:** • 0x0: Initial Hash from standard (default) • 0x1: User-defined initial Hash enabled

8.7.3.4.3 STAT

- **Name:** HMAC Status Register
- **Description:** This is a read-only register used to indicate the current status of HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x8
- **Reset Value:** 0x00000000

Table 8-534 HMAC Status Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5	DMA_TX_DONE	R	0x0	HMAC DMA transfer done or not
4	HMAC_READY	R	0x0	HMAC process status **Value:** • 0x0: Result data is not ready. • 0x1: Result data is valid.
3	KEY_VALID	R	0x0	HMAC has fetched key or not
2	DMA_TX_ERR	R WC	0x0	HMAC DMA transfer error. Write 1 to clear.
1	DMA_MSG_DONE	R	0x0	If the number of all block messages is bigger than the transfer size, it indicates all block messages are sent when this signal is set.
0	HASH_READY	R	0x0	Hash process status **Value:** • 0x0: Result data is not ready. • 0x1: Result data is valid.

8.7.3.4.4 XFER_SIZE

- **Name:** HMAC Transfer Size Register
- **Description:** This register sets a size for HMAC DMA mode transfer.
- **Base Address:** 0x40015800
- **Offset:** 0xC
- **Reset Value:** 0x00000000

Table 8-535 HMAC Transfer Size Register

Bits	Field Name	RW	Reset	Description
31:15	RSVD	R		Reserved bits
14:0	SIZE	RW	0x0	Total transfer size. Up to 32 KB • 0x003F: 1 block • 0x007F: 2 blocks • 0x00BF: 3 blocks …

Bits	Field Name	RW	Reset	Description
				• 0x7FFF: 512 blocks

8.7.3.4.5 INT

- **Name:** HMAC Interrupt Register
- **Description:** This register enables or disables all interrupts generated by the HMAC. Additionally, it reports the status of the HMAC interrupts after they have been enabled.
- **Base Address:** 0x40015800
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 8-536 HMAC Interrupt Register

Bits	Field Name	RW	Reset	Description
31:2	RSVD	R		Reserved bits
1	EN	RW	0x0	HMAC result data complete interrupt **Value:** • 0x0: Disable • 0x1: Enable
0	DONE	R WC	0x0	HMAC result data complete interrupt flag. Write 1 to clear. **Read:** • 0x0: Not interrupt • 0x1: Interrupt **Write:** • 0x0: Not effect • 0x1: Clear

8.7.3.4.6 RD_START_ADDR

- **Name:** HMAC Read Start Address Register
- **Description:** This register acts as a start address for HMAC DMA reading.
- **Base Address:** 0x40015800
- **Offset:** 0x14
- **Reset Value:** 0x00000000

Table 8-537 HMAC Read Start Address Register

Bits	Field Name	RW	Reset	Description
31:0	ADDR	RW	0x0	In DMA mode, set the read start address of transfer.

8.7.3.4.7 WR_START_ADDR

- **Name:** HMAC Write Start Address Register
- **Description:** This register acts as a start address for HMAC DMA writing.
- **Base Address:** 0x40015800
- **Offset:** 0x18
- **Reset Value:** 0x00000000

Table 8-538 HMAC Write Start Address Register

Bits	Field Name	RW	Reset	Description
31:0	ADDR	RW	0x0	In DMA mode, set the write start address of transfer.

8.7.3.4.8 USER_HASH_0

- **Name:** HMAC User Hash 0 Register
- **Description:** This register acts as a user Hash value for HASH calculation.
- **Base Address:** 0x40015800
- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 8-539 HMAC User Hash 0 Register

Bits	Field Name	RW	Reset	Description
Bits	Field Name	RW	Reset	Description
31:0	HASH_0	RW	0x0	If UHASH is selected, you can configure a user-defined Hash value by using USER_HASH_0/1/2/3/4/5/6/7. User defined Hash value[255:224]

8.7.3.4.9 USER_HASH_1

- **Name:** HMAC User Hash 1 Register
- **Description:** This register acts as a user Hash value for HASH calculation.
- **Base Address:** 0x40015800
- **Offset:** 0x24

- **Reset Value:** 0x00000000

Table 8-540 HMAC User Hash 1 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_1	RW	0x0	If UHASH is selected, you can configure a user-defined Hash value by using USER_HASH_0/1/2/3/4/5/6/7. User defined Hash value[223:192]

8.7.3.4.10 USER_HASH_2

- **Name:** HMAC User Hash 2 Register
- **Description:** This register acts as a user Hash value for HASH calculation.
- **Base Address:** 0x40015800
- **Offset:** 0x28
- **Reset Value:** 0x00000000

Table 8-541 HMAC User Hash 2 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_2	RW	0x0	If UHASH is selected, you can configure a user-defined Hash value by using USER_HASH_0/1/2/3/4/5/6/7. User defined Hash value[191:160]

8.7.3.4.11 USER_HASH_3

- **Name:** HMAC User Hash 3 Register
- **Description:** This register acts as a user Hash value for HASH calculation.
- **Base Address:** 0x40015800
- **Offset:** 0x2C
- **Reset Value:** 0x00000000

Table 8-542 HMAC User Hash 3 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_3	RW	0x0	If UHASH is selected, you can configure a user-defined Hash value by using USER_HASH_0/1/2/3/4/5/6/7. User defined Hash value[159:128]

8.7.3.4.12 USER_HASH_4

- **Name:** HMAC User Hash 4 Register
- **Description:** This register acts as a user Hash value for HASH calculation
- **Base Address:** 0x40015800
- **Offset:** 0x30
- **Reset Value:** 0x00000000

Table 8-543 HMAC User Hash 4 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_4	RW	0x0	If UHASH is selected, you can configure a user-defined Hash value with USER_HASH_0/1/2/3/4/5/6/7. User defined Hash value[127:96]

8.7.3.4.13 USER_HASH_5

- **Name:** HMAC User Hash 5 Register
- **Description:** This register acts as a user Hash value for HASH calculation.
- **Base Address:** 0x40015800
- **Offset:** 0x34
- **Reset Value:** 0x00000000

Table 8-544 HMAC User Hash 5 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_5	RW	0x0	If UHASH is selected, you can configure a user-defined Hash value by using USER_HASH_0/1/2/3/4/5/6/7. User defined Hash value[95:64]

8.7.3.4.14 USER_HASH_6

- **Name:** HMAC User Hash 6 Register
- **Description:** This register acts as a user Hash value for HASH calculation.
- **Base Address:** 0x40015800
- **Offset:** 0x38
- **Reset Value:** 0x00000000

Table 8-545 HMAC User Hash 6 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_6	RW	0x0	If UHASH is selected, you can configure a user-defined Hash value by using USER_HASH_0/1/2/3/4/5/6/7. User defined Hash value[63:32]

8.7.3.4.15 USER_HASH_7

- **Name:** HMAC User Hash 7 Register
- **Description:** This register acts as a user Hash value for HASH calculation.
- **Base Address:** 0x40015800
- **Offset:** 0x3C
- **Reset Value:** 0x00000000

Table 8-546 HMAC User Hash 7 Register

Bits	Field Name	RW	Reset	Description
31:0	HASH_7	RW	0x0	If UHASH is selected, you can configure a user-defined Hash value by using USER_HASH_0/1/2/3/4/5/6/7. User defined Hash value[31:0]

8.7.3.4.16 DATA_OUT

- **Name:** HMAC Data Output Register
- **Description:** This register is a 32-bit only read buffer for the result data from HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 8-547 HMAC Data Output Register

Bits	Field Name	RW	Reset	Description
31:0	DATA	R	0x0	In MCU mode, users can read out results with this register: When all blocks of messages are processed, 256-bit HMAC data should be read out 8 or 9 times. HMAC data can be read when hmac_ready is valid.

8.7.3.4.17 DATA_IN

- **Name:** HMAC Data Input Register

- **Description:** This register is a 32-bit only write buffer for the input data for HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x44
- **Reset Value:** 0x00000000

Table 8-548 HMAC Data Input Register

Bits	Field Name	RW	Reset	Description
Bits	Field Name	RW	Reset	Description
31:0	DATA	W	0x0	In MCU mode, users can configure input data with this register: 32-bit data should be sent 16 or 17 times.

8.7.3.4.18 KEY0

- **Name:** HMAC Key 0 Register
- **Description:** This register is a 32-bit only write buffer for the key data[255:224] for HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x48
- **Reset Value:** 0x00000000

Table 8-549 HMAC Key 0 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY0	W	0x0	Key is set as 256 bits (8 words). WORD0

8.7.3.4.19 KEY1

- **Name:** HMAC Key 1 Register
- **Description:** This register is a 32-bit only write buffer for the key data[223:192] for HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x4C
- **Reset Value:** 0x00000000

Table 8-550 HMAC Key 1 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY1	W	0x0	Key is set as 256 bits (8 words). WORD1

8.7.3.4.20 KEY2

- **Name:** HMAC Key 2 Register
- **Description:** This register is a 32-bit only write buffer for the key data[191:160] for HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x50
- **Reset Value:** 0x00000000

Table 8-551 HMAC Key 2 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY2	W	0x0	Key is set as 256 bits (8 words). WORD2

8.7.3.4.21 KEY3

- **Name:** HMAC Key 3 Register
- **Description:** This register is a 32-bit only write buffer for the key data[159:128] for HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x54
- **Reset Value:** 0x00000000

Table 8-552 HMAC Key 3 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY3	W	0x0	Key is set as 256 bits (8 words). WORD3

8.7.3.4.22 KEY4

- **Name:** HMAC Key 4 Register
- **Description:** This register is a 32-bit only write buffer for the key data[127:96] for HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x58
- **Reset Value:** 0x00000000

Table 8-553 HMAC Key 4 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY4	W	0x0	Key is set as 256 bits (8 words).

Bits	Field Name	RW	Reset	Description
				WORD4

8.7.3.4.23 KEY5

- **Name:** HMAC Key 5 Register
- **Description:** This register is a 32-bit only write buffer for the key data[95:64] for HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x5C
- **Reset Value:** 0x00000000

Table 8-554 HMAC Key 5 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY5	W	0x0	Key is set as 256 bits (8 words). WORD5

8.7.3.4.24 KEY6

- **Name:** HMAC Key 6 Register
- **Description:** This register is a 32-bit only write buffer for the key data[63:32] for HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x60
- **Reset Value:** 0x00000000

Table 8-555 HMAC Key 6 Register

Bits	Field Name	RW	Reset	Description
Bits	Field Name	RW	Reset	Description
31:0	KEY6	W	0x0	Key is set as 256 bits (8 words). WORD6

8.7.3.4.25 KEY7

- **Name:** HMAC Key 7 Register
- **Description:** This register is a 32-bit only write buffer for the key data[31:0] for HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x64
- **Reset Value:** 0x00000000

Table 8-556 HMAC Key 7 Register

Bits	Field Name	RW	Reset	Description
31:0	KEY7	W	0x0	Key is set as 256 bits (8 words). WORD7

8.7.3.4.26 KEY_ADDR

- **Name:** HMAC Key Address Register
- **Description:** This register acts as a key address for HMAC.
- **Base Address:** 0x40015800
- **Offset:** 0x68
- **Reset Value:** 0x00000000

Table 8-557 HMAC Key Address Register

Bits	Field Name	RW	Reset	Description
31:0	KEY_ADDR	RW	0x0	Indicates the HMAC key address in memory.

8.7.3.4.27 KEYPORT_MASK

- **Name:** HMAC Keyport Mask Register
- **Description:** This register acts as a mask for key from key port.
- **Base Address:** 0x40015800
- **Offset:** 0x6C
- **Reset Value:** 0x00000000

Table 8-558 HMAC Keyport Mask Register

Bits	Field Name	RW	Reset	Description
31:0	MASK	W	0x0	Mask for key from key port

8.7.4 Public Key Cryptography (PKC)

8.7.4.1 Introduction

Public Key Cryptography (PKC) algorithm, also known as asymmetric key algorithm, uses a pair of keys: a public key and a private key. The security of the private key should be ensured. The public key can be published. Data encrypted with the public key can only be decrypted with the private key, and vice versa. The public key algorithm does not need an online key server, and the key distribution protocol is simple, which greatly simplifies the key management. In addition to the encryption function, the public key algorithm provides key exchange function and digital signature function.

8.7.4.2 Main Features

- Compliance with FIPS 186-4 standard
- Supports Scalar multiplication for P-256 elliptic curve
- Supports diverse modular operations:
  - Montgomery modular multiplication for a configurable 256–2048 bits' length
  - Partial Montgomery Inversion for a configurable 256–2048 bits' length
  - Modular addition for a configurable 256–2048 bits' length
  - Modular subtraction for a configurable 256–2048 bits' length
  - Modular comparison for a configurable 256–2048 bits' length
  - Modular left-shift operation for a configurable 256–2048 bits' length
  - Big integer multiplication for a configurable 256–1024 bits' length
  - Big integer addition for a configurable 256–2048 bits' length
  - Hardware dummy multiplication
- Supports random clocking gating.
- Protects register read/write operations (registers cannot be read or written in the DISABLE module state).
- Supports interrupt and query.

8.7.4.3 Functional Description

8.7.4.3.1 Standard

Table 8-559 PKC module standard

Document Code	Name of Standard
FIPS PUB 186-4	Digital Signature Standard (DSS) (Compliant with P-256)
SEC 2	Recommended Elliptic Curve Domain Parameters

8.7.4.3.2 Module Overview

The PKC controller module completes the basic underlying modular arithmetic and the FIPS standard 256-point ECC point multiplication in the public key algorithm. GR5526 embeds the PKC module in order to realize hardware acceleration of calculation.

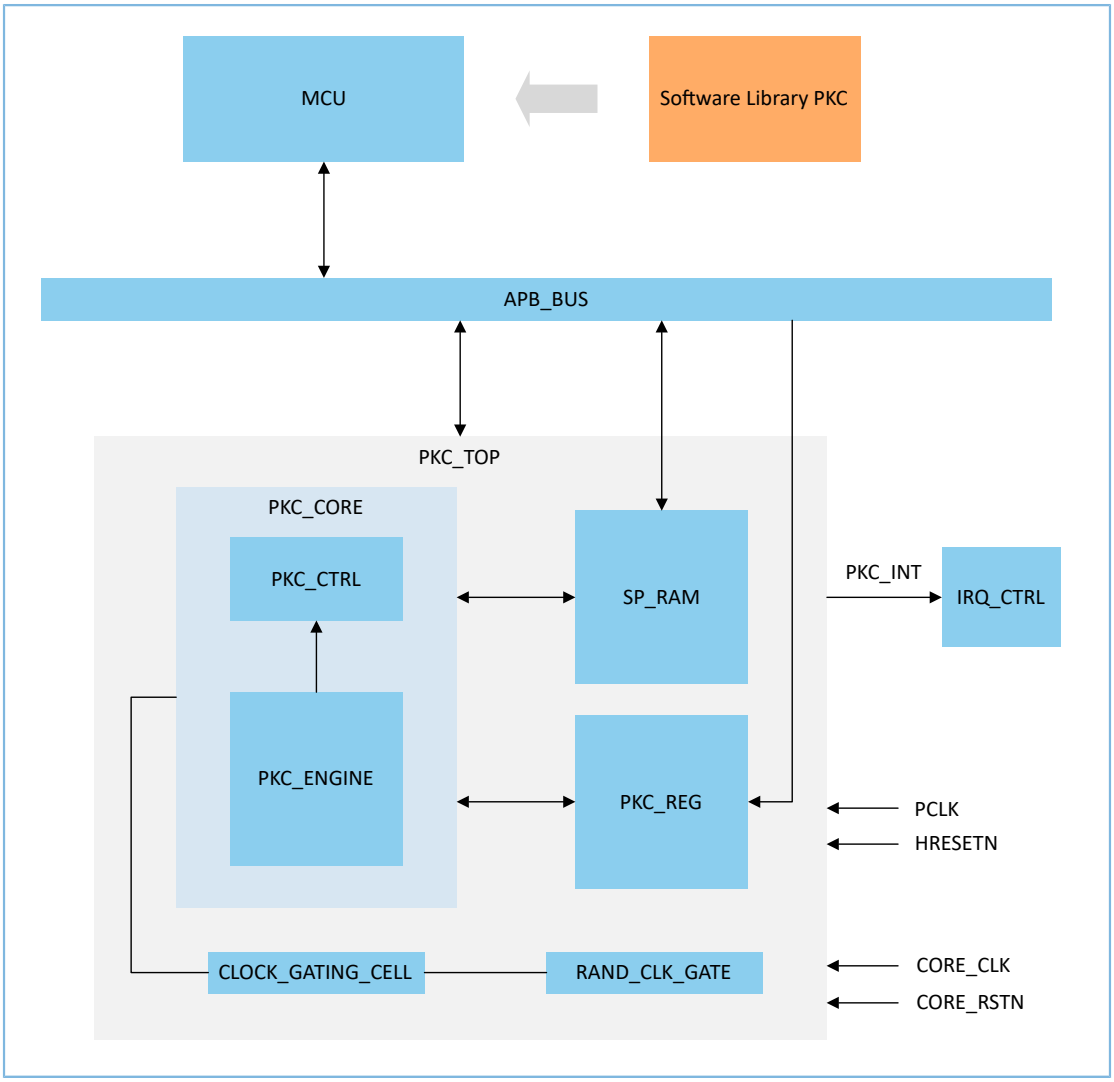

Figure 8-115 PKC module in the system

PKC_TOP includes PKC_CORE, PKC_REG, RAND_CLK_GATE, and SP_RAM. PKC_CORE includes PKC_ENGINE and PKC_CTRL sub modules, in which PKC_ENGINE completes the basic underlying modular operations (modular multiplication, modular inversion, modular addition, modular subtraction, modular comparison, modular shift, big integer multiplication, big integer addition, and dummy multiplication) in the public key algorithm.

The ECC point multiplication only supports P-256 curve, such as secp256r1 and secp256k1. The public key algorithm needs a software library which can be open source or user-defined for implementation.

8.7.4.3.3 Usage

The CTRL register is used to start and reset the PKC module. The STAT register is used to indicate the status of the PKC module. The INT_EN register is used to configure the interrupt mode, and the INT_STAT register is used to indicate the interrupt status.

- **Point Multiplication:** MCU writes the following parameters to SP_RAM that is a memory area of the PKC module. The parameters include the 256-bit random number k, order n, prime P, base point G, R2mod P, and the initial

infinity point Q0. CFG0 to CFG12 registers are used to configure the location pointers. The calculation can be started by using the CTRL register. MCU will read the result from the SP_RAM when the calculation completes. In the interrupt mode, the INT_STAT register can indicate whether the calculation completes.

- **Modular Multiplication:** MCU writes operand a, operand b, and prime p to SP_RAM. The SW_CFG7 register is used to configure the operand word length. SW_CFG0 and SW_CFG1 registers are used to configure the location pointers. The CFG13 register is used to configure the modular multiplication constant if the lowest 32-bit of prime p is not 0xFFFFFFFF. The calculation can be started by using the CTRL register. MCU will read the result from the SP_RAM when the calculation completes. In the interrupt mode, the INT_STAT register can indicate whether the calculation completes.
- **Modular Inversion:** MCU writes operand a, modulo p, x1 initial value 1, and x2 initial value 0 to SP_RAM. The SW_CFG7 register is used to configure the operand word length. SW_CFG4 and SW_CFG5 registers are used to configure the location pointers. The SW_CFG6 register is configured to save the intermediate result of modular inversion operation. The calculation can be started by using the CTRL register. MCU will read the modular inversion x1 from the SP_RAM and read the SW_CFG8 register to get the modular inversion k. In the interrupt mode, the INT_STAT register can indicate whether the calculation completes.
- **Modular Subtraction and Modular Addition:** MCU writes operand a, operand b, and prime p to SP_RAM. The SW_CFG7 register is used to configure the operand word length. SW_CFG2 and SW_CFG3 registers are used to configure the location pointers. The calculation can be started with the CTRL register. MCU will read the result from the SP_RAM when the calculation completes. In the interrupt mode, the INT_STAT register can indicate whether the calculation completes.
- **Modular Comparison and Modular Shift:** MCU writes operand a and prime p to SP_RAM. The SW_CFG7 register is used to configure the operand word length. SW_CFG0 and SW_CFG1 registers are used to configure the location pointers. The calculation can be started by using the CTRL register. MCU will read the result from the SP_RAM when the calculation completes. In the interrupt mode, the INT_STAT register can indicate whether the calculation completes.
- **Big Integer Multiplication:** MCU writes operand a and operand b to SP_RAM. The SW_CFG7 register is used to configure the operand word length. The maximum configurable length of the big number multiplication is 1024 bits. SW_CFG10 and SW_CFG11 registers are used to configure the location pointers. The calculation can be started by using the CTRL register. MCU will read the result from the SP_RAM when the calculation completes. In the interrupt mode, the INT_STAT register can indicate whether the calculation completes.
- **Big Integer Addition:** MCU writes operand a and operand b to SP_RAM. The SW_CFG7 register is used to configure the operand word length. SW_CFG11 and SW_CFG12 registers are used to configure the location pointers. The calculation can be started by using the CTRL register. MCU will read the result from the SP_RAM when the calculation completes. In the interrupt mode, the INT_STAT register can indicate whether the calculation completes.
- **Dummy Multiplication:** The SW_CFG9 register is used to configure the random seed of dummy multiplication. The dummy multiplication can be started by using the CTRL register. Subsequently, modular addition or big integer addition is performed. The result of the modular addition or the big integer addition can be obtained.

- **Random Clock Gating:** The SW_CFG13 register is used to configure the random seed of random clock gating. The random clock gating can be enabled by using the CTRL register. Subsequently, PKC operation can be performed.

8.7.4.4 Registers

8.7.4.4.1 CTRL

- **Name:** PKC Controller Register
- **Description:** This register acts as a global enable/disable for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 8-560 PKC Controller Register

Bits	Field Name	RW	Reset	Description
31:9	RSVD	R		Reserved bits
8	RST	RW	0x0	Write 0 and then write 1 to this bit to force the PKC core to reset.
7:5	RSVD	R		Reserved bits
4	SW_CTRL	RW	0x0	Only used in software mode. **Value:** • 0x0: Disable software controller. • 0x1: Enable software controller.
3:2	RSVD	R		Reserved bits
1	START	RW	0x0	Only used in hardware mode. After MCU configures all parameters; write 0 and then write 1 to this bit, and the PKC starts to work.
0	EN	RW	0x0	PKC core enable bit **Value:** • 0x0: Disable PKC. • 0x1: Enable PKC.

8.7.4.4.2 CFG0

- **Name:** PKC Configuration 0 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000

- **Offset:** 0x4
- **Reset Value:** 0x00100000

Table 8-561 PKC Configuration 0 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	R_POINT	RW	0x10	R point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	K_POINT	RW	0x0	K point in sp_ram

8.7.4.4.3 CFG1

- **Name:** PKC Configuration 1 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x8
- **Reset Value:** 0x00280020

Table 8-562 PKC Configuration 1 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	R2_POINT	RW	0x28	R^2 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	P_POINT	RW	0x20	P point in sp_ram

8.7.4.4.4 CFG2

- **Name:** PKC Configuration 2 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0xC
- **Reset Value:** 0x00380030

Table 8-563 PKC Configuration 2 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	GY_POINT	RW	0x38	Gy point in sp_ram

Bits	Field Name	RW	Reset	Description
15:9	RSVD	R		Reserved bits
8:0	GX_POINT	RW	0x30	Gx point in sp_ram

8.7.4.4.5 CFG3

- **Name:** PKC Configuration 3 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x10
- **Reset Value:** 0x00480040

Table 8-564 PKC Configuration 3 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	R0X_POINT	RW	0x48	R0x point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	GZ_POINT	RW	0x40	Gz point in sp_ram

8.7.4.4.6 CFG4

- **Name:** PKC Configuration 4 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x14
- **Reset Value:** 0x00580050

Table 8-565 PKC Configuration 4 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	R0Z_POINT	RW	0x58	R0z point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	R0Y_POINT	RW	0x50	R0y point in sp_ram

8.7.4.4.7 CFG5

- **Name:** PKC Configuration 5 Register

- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x18
- **Reset Value:** 0x00680060

Table 8-566 PKC Configuration 5 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	R1Y_POINT	RW	0x68	R1y point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	R1X_POINT	RW	0x60	R1x point in sp_ram

8.7.4.4.8 CFG6

- **Name:** PKC Configuration 6 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x1C
- **Reset Value:** 0x00780070

Table 8-567 PKC Configuration 6 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	TEMP1_POINT	RW	0x78	Tmp1 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	R1Z_POINT	RW	0x70	R1Z point in sp_ram

8.7.4.4.9 CFG7

- **Name:** PKC Configuration 7 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x20
- **Reset Value:** 0x00880080

Table 8-568 PKC Configuration 7 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	TEMP3_POINT	RW	0x88	TMP3 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	TEMP2_POINT	RW	0x80	TMP2 point in sp_ram

8.7.4.4.10 CFG8

- **Name:** PKC Configuration 8 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x24
- **Reset Value:** 0x00980090

Table 8-569 PKC Configuration 8 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	TEMP5_POINT	RW	0x98	TMP5 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	TEMP4_POINT	RW	0x90	TMP4 point in sp_ram

8.7.4.4.11 CFG9

- **Name:** PKC Configuration 9 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x28
- **Reset Value:** 0x00A800A0

Table 8-570 PKC Configuration 9 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	CONT1_POINT	RW	0xA8	Constant 1 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	TEMP6_POINT	RW	0xA0	TMP6 point in sp_ram

8.7.4.4.12 CFG10

- **Name:** PKC Configuration 10 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x2C
- **Reset Value:** 0x00B800B0

Table 8-571 PKC Configuration 10 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	X2_POINT	RW	0xB8	X2 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	X1_POINT	RW	0xB0	X1 point in sp_ram

8.7.4.4.13 CFG11

- **Name:** PKC Configuration 11 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x30
- **Reset Value:** 0x00C800C0

Table 8-572 PKC Configuration 11 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	KT_POINT	RW	0xC8	Software computes 2^256 mod p and writes to sp_ram.
15:9	RSVD	R		Reserved bits
8:0	MIT_POINT	RW	0xC0	Only used in hardware mode. MI point in sp_ram

8.7.4.4.14 CFG12

- **Name:** PKC Configuration 12 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x34

- **Reset Value:** 0x00D800D0

Table 8-573 PKC Configuration 12 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	B_POINT	RW	0xD8	Curve parameter b in Montgomery field
15:9	RSVD	R		Reserved bits
8:0	A_POINT	RW	0xD0	Curve parameter a in Montgomery field

8.7.4.4.15 CFG13

- **Name:** PKC Configuration 13 Register
- **Description:** This register acts as a configuration for PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x38
- **Reset Value:** 0x00000001

Table 8-574 PKC Configuration 13 Register

Bits	Field Name	RW	Reset	Description
31:0	CONSTQ	RW	0x1	Constant used in Montgomery multiplication

8.7.4.4.16 SW_CTRL

- **Name:** PKC Software Controller Register
- **Description:** This register acts as a controller for software operation.
- **Base Address:** 0x40014000
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 8-575 PKC Software Controller Register

Bits	Field Name	RW	Reset	Description
31:10	RSVD	R		Reserved bits
9	RCG_EN	RW	0x0	Enable random clock gating. **Value:** • 0x0: Disable • 0x1: Enable
8	DM_EN	RW	0x0	Enable dummy multiplication.

Bits	Field Name	RW	Reset	Description
				Value: • 0x0: Disable • 0x1: Enable
7	RSVD	R		Reserved bit
6:4	MODE	RW	0x0	Only used in software mode. **Value:** • 0x0: Montgomery multiplication • 0x1: Mod inversion • 0x2: Mod addition • 0x3: Mod subtraction • 0x4: Mod comparison • 0x5: Mod shift • 0x6: Big integer multiplication • 0x7: Bit integer addition
3:1	RSVD	R		Reserved bits
0	START	RW	0x0	Only used in software mode. MCU writes 0 to this bit and then writes 1 to this bit to start the PKC.

8.7.4.4.17 SW_CFG0

- **Name:** PKC Software Configuration 0 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x44
- **Reset Value:** 0x00080000

Table 8-576 PKC Software Configuration 0 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	MMB_POINT	RW	0x8	Modular multiplication B point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	MMA_POINT	RW	0x0	Modular multiplication A point in sp_ram

8.7.4.4.18 SW_CFG1

- **Name:** PKC Software Configuration 1 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x48
- **Reset Value:** 0x00180010

Table 8-577 PKC Software Configuration 1 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	MMC_POINT	RW	0x18	Modular multiplication C point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	MMP_POINT	RW	0x10	Modular multiplication P point in sp_ram

8.7.4.4.19 SW_CFG2

- **Name:** PKC Software Configuration 2 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x4C
- **Reset Value:** 0x00280020

Table 8-578 PKC Software Configuration 2 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	MASB_POINT	RW	0x28	Modular addition/subtraction B point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	MASA_POINT	RW	0x20	Modular addition/subtraction A point in sp_ram

8.7.4.4.20 SW_CFG3

- **Name:** PKC Software Configuration 3 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x50
- **Reset Value:** 0x00380030

Table 8-579 PKC Software Configuration 3 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	MASC_POINT	RW	0x38	Modular addition/subtraction C point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	MASP_POINT	RW	0x30	Modular addition/subtraction P point in sp_ram

8.7.4.4.21 SW_CFG4

- **Name:** PKC Software Configuration 4 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x54
- **Reset Value:** 0x00480040

Table 8-580 PKC Software Configuration 4 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	MIV_POINT	RW	0x48	Modular inversion V point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	MIU_POINT	RW	0x40	Modular inversion U point in sp_ram

8.7.4.4.22 SW_CFG5

- **Name:** PKC Software Configuration 5 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x58
- **Reset Value:** 0x00580050

Table 8-581 PKC Software Configuration 5 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	MIX2_POINT	RW	0x58	Modular inversion X2 point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	MIT_POINT	RW	0x50	Modular inversion X1 point in sp_ram

8.7.4.4.23 SW_CFG6

- **Name:** PKC Software Configuration 6 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x5C
- **Reset Value:** 0x00000060

Table 8-582 PKC Software Configuration 6 Register

Bits	Field Name	RW	Reset	Description
31:9	RSVD	R		Reserved bits
8:0	MITP	RW	0x60	Modular inversion TEMP point in sp_ram

8.7.4.4.24 SW_CFG7

- **Name:** PKC Software Configuration 7 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x60
- **Reset Value:** 0x00000007

Table 8-583 PKC Software Configuration 7 Register

Bits	Field Name	RW	Reset	Description
31:6	RSVD	R		Reserved bits
5:0	LEN	RW	0x7	Operator word length configuration in software mode: • 0x07: 256 bits • 0x08: 288 bits … • 0x3F: 2048 bits In hardware mode, this register is fixed to 0x07.

8.7.4.4.25 SW_CFG8

- **Name:** PKC Software Configuration 8 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000

- **Offset:** 0x64
- **Reset Value:** 0x00000000

Table 8-584 PKC Software Configuration 8 Register

Bits	Field Name	RW	Reset	Description
31:13	RSVD	R		Reserved bits
12:0	MIK_OUT	R	0x0	K out in modular inversion operation

8.7.4.4.26 SW_CFG9

- **Name:** PKC Software Configuration 9 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x68
- **Reset Value:** 0xAABBCCDD

Table 8-585 PKC Software Configuration 9 Register

Bits	Field Name	RW	Reset	Description
31:0	RDMS	RW	0xAABBCCDD	Random dummy multiplication seed

8.7.4.4.27 SW_CFG10

- **Name:** PKC Software Configuration 10 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x6C
- **Reset Value:** 0x00780070

Table 8-586 PKC Software Configuration 10 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	BMB_POINT	RW	0x78	Big multiplication B point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	BMA_POINT	RW	0x70	Big multiplication A point in sp_ram

8.7.4.4.28 SW_CFG11

- **Name:** PKC Software Configuration 11 Register

- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x70
- **Reset Value:** 0x00880080

Table 8-587 PKC Software Configuration 11 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	BAA_POINT	RW	0x88	Big addition A point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	BMC_POINT	RW	0x80	Big multiplication C point in sp_ram

8.7.4.4.29 SW_CFG12

- **Name:** PKC Software Configuration 12 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x74
- **Reset Value:** 0x00980090

Table 8-588 PKC Software Configuration 12 Register

Bits	Field Name	RW	Reset	Description
31:25	RSVD	R		Reserved bits
24:16	BAC_POINT	RW	0x98	Big addition C point in sp_ram
15:9	RSVD	R		Reserved bits
8:0	BAB_POINT	RW	0x90	Big addition B point in sp_ram

8.7.4.4.30 SW_CFG13

- **Name:** PKC Software Configuration 13 Register
- **Description:** This register acts as a configuration for PKC in software mode.
- **Base Address:** 0x40014000
- **Offset:** 0x78
- **Reset Value:** 0x11223344

Table 8-589 PKC Software Configuration 13 Register

Bits	Field Name	RW	Reset	Description
31:0	RCG_SEED	RW	0x11223344	Random clock gating seed

8.7.4.4.31 INT_STAT

- **Name:** PKC Interrupt Status Register
- **Description:** This is a read-only register used to indicate the interrupt status of PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x80
- **Reset Value:** 0x00000000

Table 8-590 PKC Interrupt Status Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	BIAO_INT_FLAG	R WC	0x0	PKC big integer add overflow flag. Write 1 to clear.
1	ERR_INT_EN	R WC	0x0	PKC error interrupt flag. Write 1 to clear.
0	CPLT_INT_FLAG	R WC	0x0	PKC complete interrupt flag. Write 1 to clear.

8.7.4.4.32 INT_EN

- **Name:** PKC Interrupt Enable Register
- **Description:** This register enables or disables all interrupts generated by the PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x84
- **Reset Value:** 0x00000000

Table 8-591 PKC Interrupt Enable Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2	BIAO_INT_EN	RW	0x0	PKC big integer add overflow interrupt enable
1	ERR_INT_EN	RW	0x0	PKC error interrupt enable
0	CPLT_INT_EN	RW	0x0	PKC complete interrupt enable

8.7.4.4.33 STAT

- **Name:** PKC Status Register
- **Description:** This is a read-only register used to indicate the status of PKC.
- **Base Address:** 0x40014000
- **Offset:** 0x88
- **Reset Value:** 0x00000000

Table 8-592 PKC Status Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	BUSY	R	0x0	PKC busy status

8.7.5 True Random Number Generator (TRNG)

8.7.5.1 Introduction

The True Random Number Generator (TRNG) generates random numbers used in encryption and decryption. Post-processing is added to the TRNG module to ensure the quality of random number and amend the bias of TRNG.

The goal of true random data generated by TRNG is to pass the NIST SP 800-22 standard test suite.

8.7.5.2 Main Features

- Supports true and pseudo random number generation.
- Random number quality can pass the NIST SP 800-22 standard test suite.
- Supports multiple post-processing mechanisms, including bit-skipping, bit-counting, and von-Neumann.
- Supports multiple combinations of random numbers.
- Supports both interrupt and query modes.

8.7.5.3 Functional Description

8.7.5.3.1 Standard

Table 8-593 TRNG module standard

Document Code	Name of Standard
NIST SP 800-22	A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications

8.7.5.3.2 Module Overview

The TRNG module supports pseudo random number generation and true random number generation. Pseudo random numbers are generated by a linear feedback shift register (LFSR). True random numbers are generated by using a feedforward ring oscillator (FRO) as the entropy source.

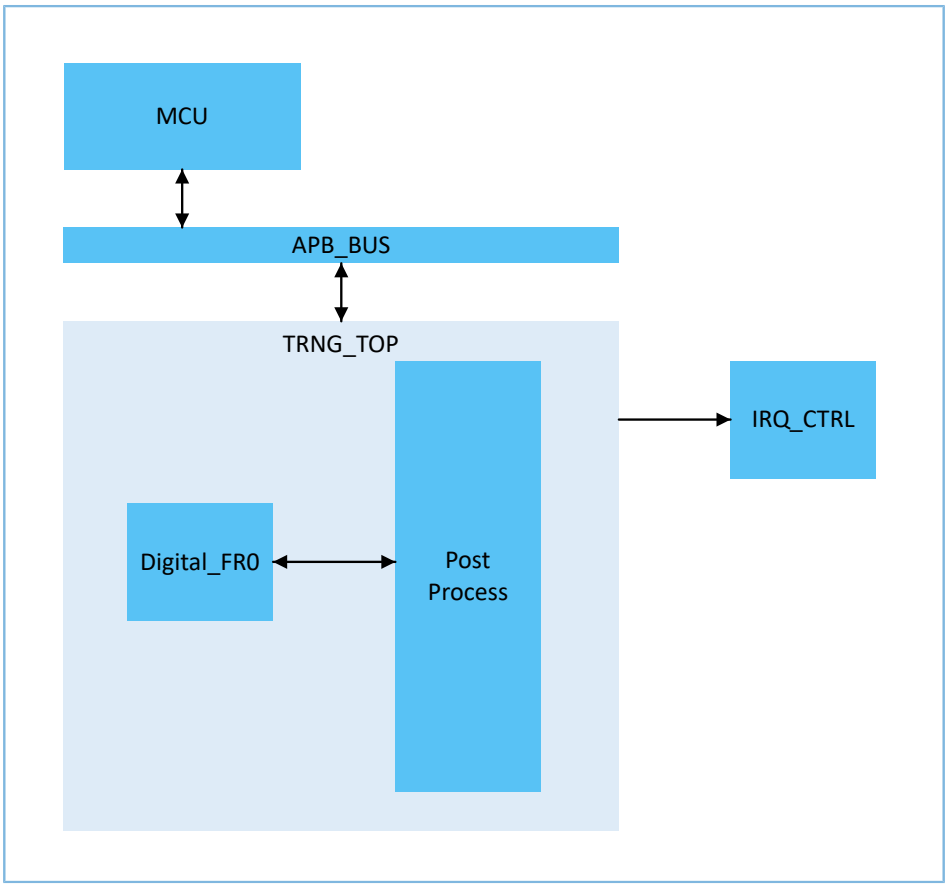

Figure 8-116 RNG module in the system

Digital_FRO is the digital true random number generator, based on the FRO scheme.

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

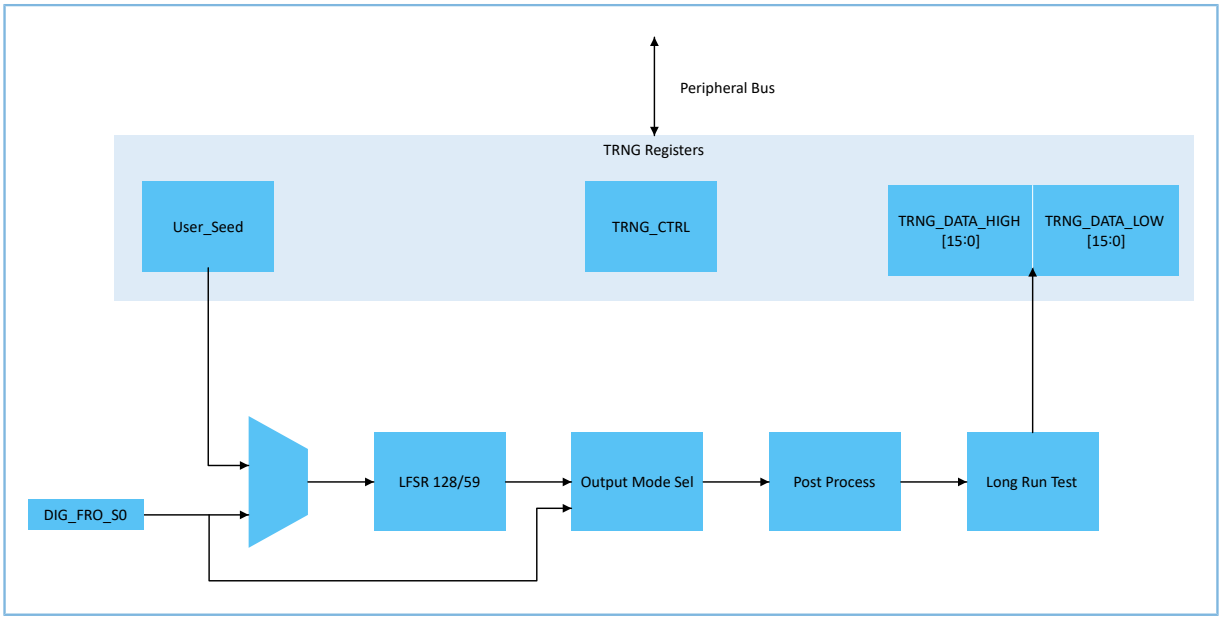

Figure 8-117 TRNG post-process module

The work mode of the LFSR can be configured as 128 or 59 polynomial which is implemented by Galois structure. The LFSR seed can be configured from the FRO or in parallel be written in user-configured seed by using the user_seed register.

The TRNG module supports multiple output modes, including LFSR direct output, LFSR XOR TRNG, LFSR and TRNG cyclic sampling, LFSR and TRNG cyclic sampling and parity generation, and the FRO direct output. Since only the FRO is used as the true random number source, the results of LFSR direct output, LFSR XOR TRNG, as well as LFSR and TRNG cyclic sampling will be the same when the LFSR seed is configured by users.

The TRNG module supports multiple post-processing mechanisms, including bit-skipping, bit-counting, and von-Neumann.

8.7.5.3.3 Usage

The CFG register can be used to configure relevant parameters for the TRNG module. CFG.OUT_MODE can be used to configure the output mode. CFG.P_MODE_CFG can be used to configure the post-process mode. CFG.LFSR_MODE_CFG can be used to configure the 59-bit or 128-bit LFSR operation mode. CFG.LFSR_SEED_CFG can be used to configure the LFSR seed source. CFG.HW_INT_EN can set the TRNG module into interrupt mode.

The FRO acts as the entropy source, so that CFG.FRO_EN should be enabled in order to generate true random numbers. It can directly output or be set as the LFSR seed.

The TRNG module also supports user-defined LFSR seeds. In the 59-bit LFSR operation mode, the USER_SEED register needs to be written 4 consecutive times to configure the seed. In the 128-bit LFSR operation mode, the USER_SEED register needs to be written 8 consecutive times to configure the seed. If the output mode is set to LFSR directly output, the random number will be pseudo random number.

The TRNG module can be started by using the CTRL register. The STAT register is used to indicate the current status of the TRNG module. The generated random number can be read out from the DATA register.

8.7.5.4 Registers

8.7.5.4.1 CTRL

- **Name:** TRNG Controller Register
- **Description:** This register acts as a global enable/disable for TRNG.
- **Base Address:** 0x40017800
- **Offset:** 0x0
- **Reset Value:** 0x00000000

Table 8-594 TRNG Controller Register

Bits	Name	Access	Reset Value	Description
31:1	RSVD	R		Reserved bits
0	RUN	RW	0x0	TRNG working enable signal, valid at HIGH • 0x0: Disable TRNG module. • 0x1: TRNG starts working to execute an operation of reading random number.

8.7.5.4.2 STAT

- **Name:** TRNG Status Register
- **Description:** This is a read-only register used to indicate the current status of TRNG.
- **Base Address:** 0x40017800
- **Offset:** 0x4
- **Reset Value:** 0x00000000

Table 8-595 TRNG Status Register

Bits	Name	Access	Reset Value	Description
31:1	RSVD	R		Reserved bits
0	READY	R WC	0x0	TRNG status flag bit, queried by CPU. **Value:** • 0x0: TRNG output data is not valid. • 0x1: TRNG output data is valid. This register is automatically cleared by hardware after READY is set HIGH. This register is automatically cleared by hardware after data reads.

8.7.5.4.3 DATA

- **Name:** TRNG Data Register
- **Description:** This register is a 32-bit read buffer for the random numbers.
- **Base Address:** 0x40017800
- **Offset:** 0x8
- **Reset Value:** 0x00000000

Table 8-596 TRNG Data Register

Bits	Name	Access	Reset Value	Description
31:0	DATA	R	0x0	TRNG data. This register is automatically cleared by hardware after data reads.

8.7.5.4.4 MCULOCK

Not used in GR5526.

8.7.5.4.5 LONG_RUN_STAT

- **Name:** TRNG Long Run Status Register
- **Description:** This is a read-only register used to indicate the current status of long run mode.
- **Base Address:** 0x40017800
- **Offset:** 0x10
- **Reset Value:** 0x00000000

Table 8-597 TRNG Long Run Status Register

Bits	Name	Access	Reset Value	Description
31:9	RSVD	R		Reserved bits
8:1	LONG_RUN_COUNT	R	0x0	Counts of TRNG long run test. This value is incremented by 1 for every long run detected by TRNG. After TRNG is disabled or reset, the register is cleared.
0	LONG_RUN_FLAG	R	0x0	Flag of TRNG long run test **Value:** • 0x0: Long run never occurs. • 0x1: Long run was occurred in random number sequence.

Bits	Name	Access	Reset Value	Description
				After TRNG is disabled or reset, this register is cleared.

8.7.5.4.6 CFG

- **Name:** TRNG Configuration Register
- **Description:** This register acts as a configuration for TRNG.
- **Base Address:** 0x40017800
- **Offset:** 0x14
- **Reset Value:** 0x0000B246

Table 8-598 TRNG Configuration Register

Bits	Name	Access	Reset Value	Description
31:16	RSVD	R		Reserved bits
15	FRO_EN	RW	0x1	Ring oscillator TRNG enable signal **Value:** • 0x0: Disable • 0x1: Enable
14	LFS_EN	RW	0x0	Low frequency sampling TRNG enable signal **Value:** • 0x0: Disable • 0x1: Enable Reserved in GR5526.
13	HW_INT_EN	RW	0x1	TRNG hardware interrupt enable bit. **Value:** • 0x0: Disable • 0x1: Enable
12:10	LFSR_SEED_CFG	RW	0x4	LFSR seed configuration mode. Select a source for LFSR seed. **Value:** • 0x0: LFSR seed is from the switching current s0. • 0x1: LFSR seed is from the switching current s1. • 0x2: LFSR seed is from the switching current s2. • 0x3: LFSR seed is from the switching current s3. • 0x4: LFSR seed is from the oscillator s0.

Bits	Name	Access	Reset Value	Description
				• 0x5: LFSR seed is from the low frequency sampling s0. • 0x6: LFSR seed is configured by users. • 0x7: Random numbers do not use LFSR and completely use true random mode. Value is limited to 0x4 or 0x6 in GR5526.
9	LFSR_MODE_CFG	RW	0x1	LFSR configuration mode: • 0x0: 59-bit LFSR • 0x1: 128-bit LFSR
8:7	P_MODE_CFG	RW	0x0	TRNG post-process configuration **Value:** • 0x0: No post-process • 0x1: Bit skipping • 0x2: Bit counting • 0x3: Von-Neumann
6:4	LFSR_XOR_SEL	RW	0x4	In output mode 8, select TRNG with LFSR XOR. 0x4: LFSR ⊕ Ring Oscillator s0 **Note:** ⊕ means XOR. Value is limited to 0x4 in GR5526.
3:0	OUT_MODE	RW	0x6	Select a TRNG output mode. **Value:** • 0x4: Digital TRNG direct output, ring oscillator s0 • 0x6: LFSR and TRNG cyclic sampling and parity generation • 0x7: LFSR and TRNG cyclic sampling • 0x8: LFSR ⊕ TRNG • 0x9: LFSR direct output **Note:** ⊕ means XOR. Value is limited to 0x4, 0x6, 0x7, 0x8, or 0x9 in GR5526.

8.7.5.4.7 SRC_CFG

- **Name:** TRNG Source Configuration Register
- **Description:** This register acts as a configuration for source of TRNG.

- **Base Address:** 0x40017800
- **Offset:** 0x18
- **Reset Value:** 0x00007864

Table 8-599 TRNG Source Configuration Register

Bits	Name	Access	Reset Value	Description
31:15	RSVD	R		Reserved bits
14:11	FRO_CHAIN_SEL	RW	0xF	Selection of Chain4–Chain7 for FRO **Value:** • 0x0: Select long ring. • 0x1: Select short ring.
10:8	RSVD	R		Reserved bits
7:0	WAIT_TIME	RW	0x64	The waiting time during which the TRNG input reaches stable. Default value is 100 time cycles.

8.7.5.4.8 FRO_CFG

- **Name:** TRNG FRO Configuration Register
- **Description:** This register acts as a configuration for FRO.
- **Base Address:** 0x40017800
- **Offset:** 0x1C
- **Reset Value:** 0x0000FFFF

Table 8-600 TRNG FRO Configuration Register

Bits	Name	Access	Reset Value	Description
31:16	RSVD	R		Reserved bits
15:8	TEST_IN	RW	0xFF	Test input signal of each chain for FRO. The default value is 1.
7:0	CHAIN_EN	RW	0xFF	Enable signal of each chain in ring oscillator module. It is valid when FROEN is high.

8.7.5.4.9 USER_SEED

- **Name:** TRNG User Seed Register
- **Description:** This register acts as a user seed value for TRNG.
- **Base Address:** 0x40017800

- **Offset:** 0x20
- **Reset Value:** 0x00000000

Table 8-601 TRNG User Seed Register

Bits	Name	Access	Reset Value	Description
31:16	RSVD	R		Reserved bits
15:0	USER_SEED	W	0x0	RNG seed configured by users. Write four times to user_seed[58:0] to configure a 59-bit random number. Write eight times to user_seed[127:0] to configure a 128-bit random number.

8.7.5.4.10 LONG_RUN_CFG

- **Name:** TRNG Long Run Configuration Register
- **Description:** This register acts as a configuration for long run mode.
- **Base Address:** 0x40017800
- **Offset:** 0x24
- **Reset Value:** 0x00000035

Table 8-602 TRNG Long Run Configuration Register

Bits	Name	Access	Reset Value	Description
31:6	RSVD	R		Reserved bits
5:1	LONG_RUN_THD	RW	0x1A	Threshold configuration of RNG long run test. The run which lasts in a longer or an equal period than this value will be detected. Default value is 26 and the maximum value is 31.
0	THD_LR_TEST	RW	0x1	RNG long run test enable bit **Value:** • 0x0: Disable • 0x1: Enable

8.8 Analog Interfaces

8.8.1 13-Bit SADC

8.8.1.1 Introduction

The Sense Analog-to-Digital Converter (SADC) is an ADC based on successive approximation registers (SARs). It is an 8-sensing-channel 13-bit 1 Msps ADC that can be used for single ended or for differential measurements. The SADC supports DMA to allow for large sample collection without overloading the MCU.

8.8.1.2 Main Features

- 13-bit analog-to-digital conversion
- 8-channel (MSIO_0–MSIO_7) analog-to-digital conversion
- Up to 16 MHz ADC clock frequency
- Up to 1 Msps conversion rate
- Programmable ADC sampling clock
- Single-ended or differential measurements
- Internal temperature and Vbattery sensors
- Ability to capture ADC samples using DMA, unburdening the MCU.
- Programmable internal reference

8.8.1.3 Block Diagram

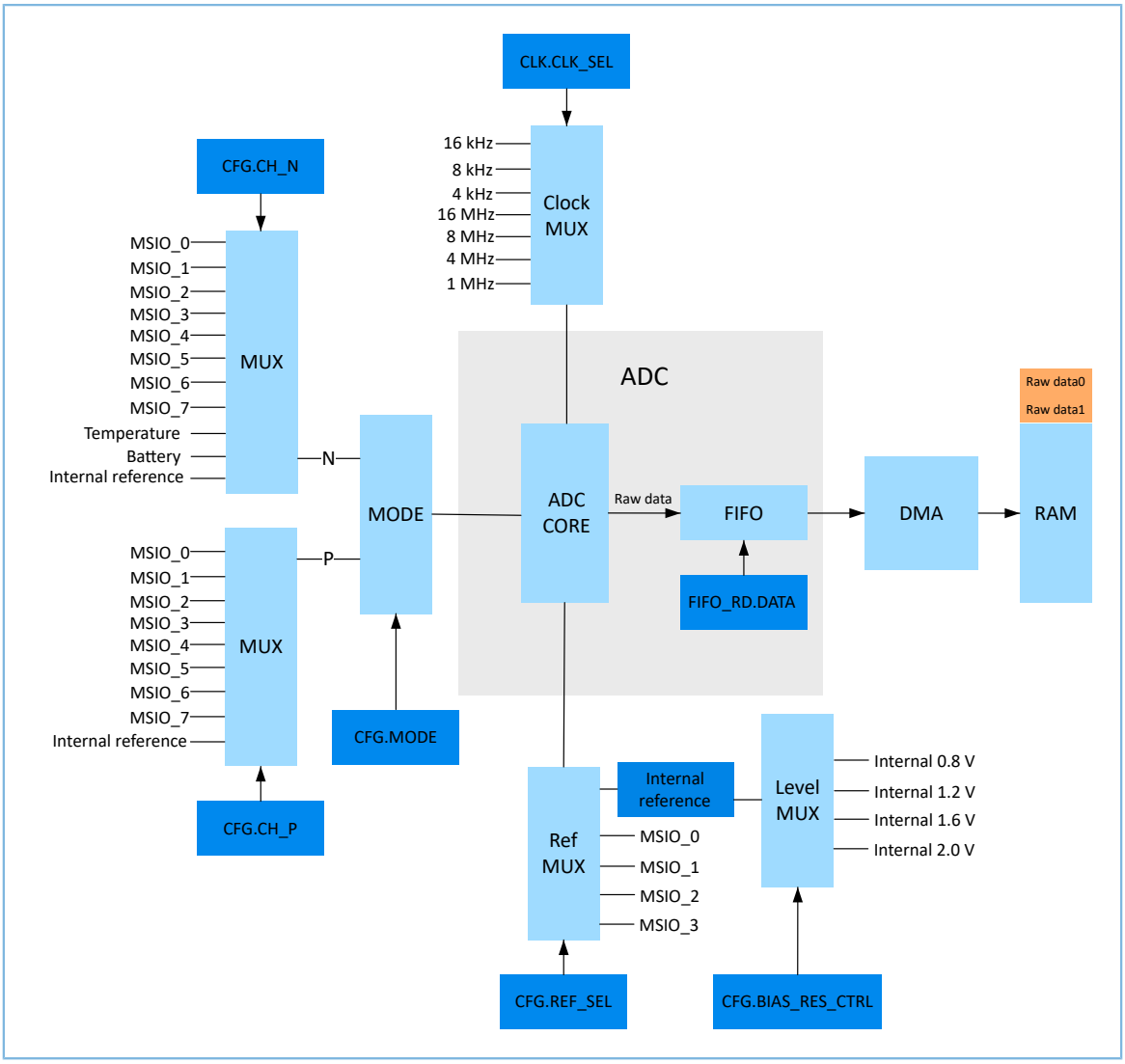

Figure 8-118 SADC block diagram

8.8.1.4 Resource Sharing

ADC and Comp can share the input channel from MSIO, but MSIO selected for Comp cannot be used at the same time for ADC sampling. The selection and configuration of IO can be found in the Pin Mux and I/O section.

8.8.1.5 Functional Description

The SADC can sample one of 8 analog sources at a given time, using an input mux. Another two of the channel_N are used internally for VBAT and temperature monitoring for adjusting the analog and RF circuitry configurations for optimum performance. SADC can be reconfigured to perform single-ended or differential measurements through CFG.MODE register. The reference voltage can be internally generated or supplied as one of the external inputs and can be selected through CFG.REF_SEL register. The external reference source can be input through MSIO_0–MSIO_3, and the input voltage must not exceed one half of the supply voltage. The internal reference voltage can be programmed as 0.8 V, 1.2 V, 1.6 V, and 2.0 V through CFG.BIAS_RES_CTRL register. Affected by ambient temperature and chip manufacturing process, the internal reference voltage will have a maximum error of 20%. For any specific

reference value (Vref), the maximum ADC input is supposed to be (2 x Vref). As shown in the following figure, the Iref comes from the relatively constant current generated by the chip's bandgap module and will not be affected by other modules; it will generate different Vref levels by resistive voltage division.

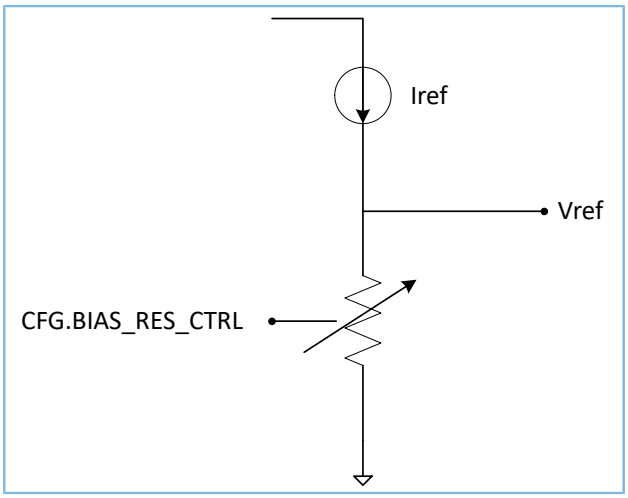

Figure 8-119 Set of ADC internal Vref control level

The ADC sampling clock can be configured as 16 kHz, 8 kHz, 4 kHz, 16 MHz, 8 MHz, 4 MHz, or 1 MHz through CLK.CLK_SEL register. The output value of the ADC will be stored into the FIFO, and users can transfer the data to RAM through DMA or MCU.

8.8.1.5.1 SADC Raw Data Stream

- Pairs of successive 13-bit samples from SADC are combined into 32-bit words by aligning the 16-bit samples. 32-bit words are stored into an SADC FIFO of 64 entries, where each entry is 32-bit long and would hold 2 samples.
- When the SADC FIFO entry count reaches the 6-bit threshold value specified in the FIFO_CTRL.THD register, the samples stored in the FIFO are automatically streamed to the SRAM by using DMA controller. (DMA0 assigned to SADC must be programmed accordingly).
- The SADC FIFO status (not-empty signal and the entry count) can be polled in the FIFO_STAT register.
- The SADC FIFO can also be “popped” by one entry, by reading the FIFO_RD register.

8.8.1.5.2 SADC Sample Operation Mode

The SADC supports two sample operation modes: single-end measurement mode and differential measurement mode. Users can choose either of the modes by programming MODE bit of CFG.MODE register.

- In single-end measurement mode, the input signal should be assigned to channel N and the SADC will only measure the input signal of channel N. In this mode, larger voltage input indicates smaller raw data output. The input signal of channel N can be selected as MSIO_0–MSIO_7, temperature sensor, battery voltage, or internal reference voltage by configuring the CFG.CH_N register. The raw data of the output is the measurement from N channel to GND and the allowed range of channel N is 0 to 2*Vref.

- In differential measurement mode, the input signals should be assigned to channel P and channel N and the SADC will measure the input signal of channel P minus channel N. The input signal of channel N is the same with that in single-end measurement mode, and the input signal of channel P can be selected as MSIO_0–MSIO_7 or internal reference voltage by configuring the CFG.CH_P register. The raw data of the output is the measurement of difference voltage between channel P to channel N and the allowed range of channel P minus channel N is -2*Vref to 2*Vref.

When users want to get the temperature sensor voltage, battery voltage, or internal reference voltage, only single-ended measurement mode can be selected.

8.8.1.5.3 SADC Clock and Sampling Rate

- The SADC supports both fast sampling and slow sampling. If you need fast sampling, you can select 1M, 4M, 8M, or 16M ADC clock frequency. If you need slow sampling, you can select 16K, 8K, or 4K ADC clock frequency. The SADC clock can be selected by configuring CLK.CLK_SEL register. The source of the SADC clock is shown in the figure below.

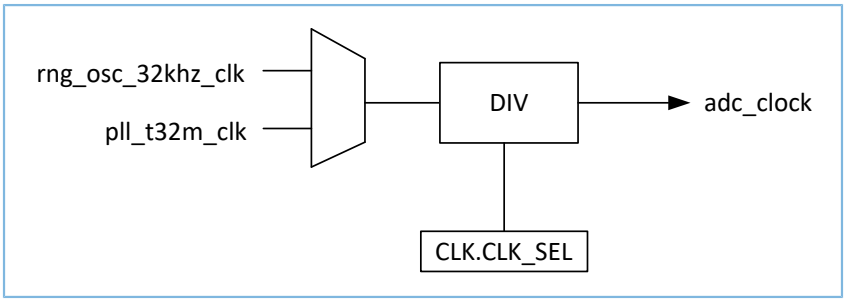

Figure 8-120 Source of SADC clock

- It takes 2 clocks for the SADC to sample and hold data, and then 13 clocks to convert analog semaphore to 13-bit binary digits, totaling 15 clocks to generate one code. Therefore, the sampling rate of the SADC is 1/15 of the ADC clock frequency. For example, a 1 MHz ADC clock has a sampling rate of 66.7 ksps.
- The ADC sampling capacitor is 2 pF, which is affected by the RC circuit. If the external impedance is larger, the charging time of the capacitor will be longer. When a higher SADC sampling rate is set, it is likely to get abnormal raw data during sampling, because the capacitor is not fully charged. This can be solved by reducing the external impedance or reducing the SADC sampling rate.

8.8.1.5.4 Considerations for Using SADC

When using an internal reference source, it is necessary for users to ensure the stability of the reference source. The internal reference voltage is divided by the chip power supply voltage, so the reference source voltage of different levels has certain requirements for the chip power supply voltage. Insufficient chip supply voltage will lead to great ADC measurement error. The chip supply voltage range corresponding to the reference sources of different gears is as follows:

Table 8-603 Chip supply voltage reference range corresponding to different reference source gears

Internal Reference	Chip Supply Voltage
0.8 V	1.8 V to 4.35 V
1.2 V	2.6 V to 4.35 V
1.6 V	3.2 V to 4.35 V
2.0 V	4.0 V to 4.35 V

For the first data acquisition after SADC initialization, the SADC module needs to spend a cycle to clear the previous sampling parameters and another cycle to make the new sampling parameters effective. Therefore, when starting the SADC for the first time, the results of the first 2 to 3 pieces of 13-bit data may be unstable. It is recommended to filter out the first 2 to 3 pieces of unstable 13-bit data after SADC starts working.

8.8.1.6 Measuring MSIO Input

Due to the offset of the ADC itself, the original value of 0–8191 cannot accurately correspond to the voltage of 0–2*Vref, so the ADC of each chip needs to be calibrated. Usually, the slope and offset of the straight line are obtained by the method of linear fitting, and the current voltage can be converted by substituting the raw data.

- In the final test stage, the **slope_int_0p8** and **offset_int_0p8** of the 0.8 V internal reference voltage are obtained by measuring the two input voltages and their corresponding raw data, and the **slope_int_0p8** and **offset_int_0p8** are stored in the eFuse (refer to Section 3.6.2 eFuse). You can use the following formula to convert the raw data into the current voltage value.

  $$cur_voltage = (raw_data - offset_int_0p8) / slope_int_0p8 \quad (30)$$

  Similarly, eFuse also stores slope_int_1p2, offset_int_1p2 of the 1.2 V internal reference voltage, and slope_int_1p6, offset_int_1p6 of the 1.6 V internal reference voltage. When the internal 1.2 V reference voltage is selected, the raw data can be converted into the current voltage value by the following formula:

  $$cur_voltage = (raw_data - offset_int_1p2) / slope_int_1p2 \quad (31)$$

  When the internal 1.6 V reference voltage is selected, the raw data can be converted into the current voltage value by the following formula:

  $$cur_voltage = (raw_data - offset_int_lp6) / slope_int_lp6 \quad (32)$$

- The **slope_ext_1p0** and **offset_ext_1p0** of the external 1.0 V reference source are also stored in the eFuse. When the external reference voltage is selected, you only need to pay attention to the scaling of the **slope_ext_1p0**. The conversion formula is as follows:

  $$cur_voltage = (raw_data - offset_ext_1p0) / (slope_ext_1p0 * 1.0 / external_reference_voltage) \quad (33)$$

The above conversion formulas are used when the single-ended mode is selected. If the differential mode is selected, the slope here will be reduced to one-half.

8.8.1.7 Measuring Battery Voltage

There is a voltage divider resistor inside the GR5526 to divide the voltage of the battery as VBATL/4. Users need to configure the CFG.CH_N register as 0xE (battery sensor as input), and select the internal reference voltage of 0.8 V.

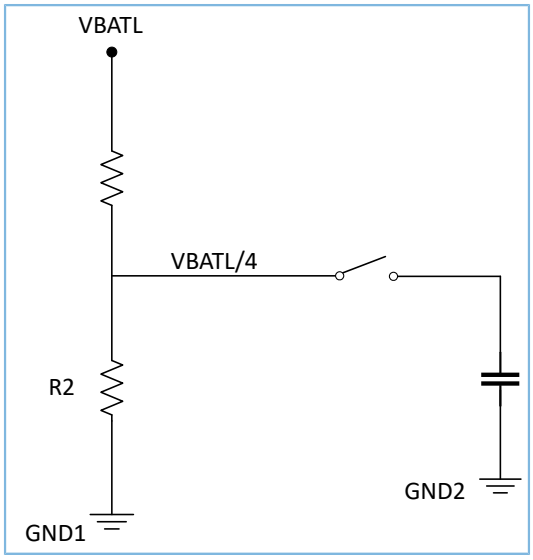

Figure 8-121 Principle of VBATL measurement

Therefore, users can obtain the raw data of the current voltage, and then convert the raw data into the actual battery voltage value through the following formula:

$$bat_voltage = [(raw_data - offset_int_0p8) / \overset{\text{Cadc}}{slope_int_0p8}] * 3.854 \quad (34)$$

Where, slope_int_0p8 and offset_int_0p8 are calibration offset and slope when 0.8 V internal reference voltage is selected; 3.854 represents the divider ratio of the resistor to the battery voltage. With the change of temperature, this divider ratio will be slightly different, ranging from 3.85 to 3.855; the typical value is 3.854.

8.8.1.8 Measuring Internal Temperature

GR5526 has a temperature sensor module inside, and the measurement range is from -40°C to 80°C. The temperature sensor converts the internal temperature of the chip into a voltage value as Vtemp and outputs it to the ADC for sample and conversion. After temperature calibration, the ADC converts the raw data into an actual temperature value.

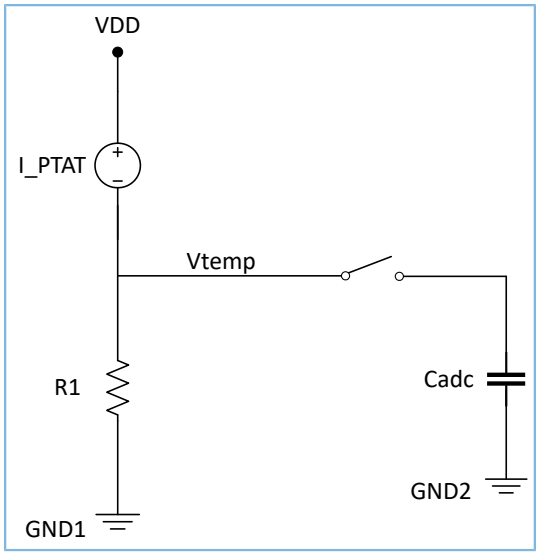

Figure 8-122 Principle of temperature measurement

If users want to measure the internal temperature of the chip, it is necessary to configure the CFG.CH_N register as 0xD (temperature sensor as input), and select the internal reference voltage of 0.8 V. The calibration formula is as follows:

$$temperature = [(raw_data - temp_25_offset) / slope_int_0p8 / (-0.00175)] + temp_25 \quad (35)$$

Where, temp_25 represents an ambient temperature of 25°C; temp_25_offset represents ADC raw data at 25°C ambient temperature; slope_int_0p8 represents the calibrated slope value of the internal 0.8 V reference voltage; -0.00175 represents the amplification factor for the slope when used to measure temperature. In general, the output voltage decreases by a typical value of 1.863 mV for every 1 degree increase in temperature. The typical value of the voltage is 807.5 mV at -40°C and 583.9 mV at 80°C.

8.8.1.9 SADC Manual Calibration Method

Although we have provided some calibration values stored in the eFuse at 3.3 V supply and normal temperature, these calibration values may have errors at other ambient temperatures. Users can use the following methods to recalibrate the conversion relationship between raw data and actual temperature or voltage.

- Single-ended mode calibration with internal reference source
  1. Select internal reference source by configuring the CFG.REF_SEL register, select internal reference level by configuring the CFG.BAIS_RES_CTRL register, and select single-ended mode by configuring the CFG.MODE register to initialize ADC.
  2. Use a noise-free power supply to connect to the N terminal by configuring the CFG.CH_N register to start ADC sampling.
  3. Sample 1024 pieces of data and remove the maximum and minimum values to obtain the average of 1022 numbers.
  4. At this time, the average value is recorded as x0, and the input voltage is recorded as y0.

5. Collect another voltage value using the above method, record the average code value as x1, and record the input voltage value as y1.
6. Use the formula $y = (x - offset) / slope$, and bring in x0, y0, x1, y1 to obtain the offset and slope.
7. The raw data can be converted into the actual voltage using the following formula:

$$actual_voltage = (raw_data - offset) / slope \tag{36}$$

- Differential-ended mode calibration with internal reference source

1. Select internal reference source by configuring the CFG.REF_SEL register, select internal reference level by configuring the CFG.BAIS_RES_CTRL register, and select differential-ended mode by configuring the CFG.MODE register to initialize ADC.
2. Use two noise-free power supplies to connect to the P terminal and N terminal by configuring the CFG.CH_P register and the CFG.CH_N register to start ADC sampling.
3. Sample 1024 pieces of data and remove the maximum and minimum values to obtain the average of 1022 numbers.
4. At this time, the average value is recorded as x0, and the value of the P terminal voltage minus the N terminal voltage is recorded as y0.
5. Collect another voltage value using the above method, record the average code value as x1, and record the input voltage difference as y1.
6. Use the formula $y = (x - offset) / slope$, and bring in x0, y0, x1, y1 to obtain the offset and slope.
7. The raw data can be converted into the actual voltage using the following formula:

$$actual_voltage = (raw_data - offset) / slope \tag{37}$$

- Single-ended mode calibration with external reference source

1. Select external reference source by configuring the CFG.REF_SEL register, and select single-ended mode by configuring the CFG.MODE register to initialize ADC.
2. Use a noise-free power supply to connect to the N terminal by configuring the CFG.CH_N register to start ADC sampling.
3. Use a noise-free power supply to connect the ADC external input pin, and record the external input as z at this time (note that the external input is only connected from MSIO_0–MSIO_3).
4. Sample 1024 pieces of data and remove the maximum and minimum values to obtain the average of 1022 numbers.
5. At this time, the average value is recorded as x0, and the input voltage is recorded as y0.
6. Collect another voltage value using the above method, record the average code value as x1, and record the input voltage value as y1.
7. Use the formula $y = (x - offset) / slope$, and bring in x0, y0, x1, y1 to obtain the offset and slope.

8. The raw_data can be converted into the actual voltage using the following formula:

$$actual_voltage = (raw_data - offset)/(slope * z / vref) \quad (38)$$

where vref represents the voltage value of the external input as this time.

- Differential-ended mode calibration with external reference source

1. Select external reference source by configuring the CFG.REF_SEL register, and select differential-ended mode by configuring the CFG.MODE register to initialize ADC.
2. Use two noise-free power supplies to connect to the P terminal and N terminal by configuring the CFG.CH_P register and the CFG.CH_N register to start ADC sampling.
3. Use a noise-free power supply to connect the ADC external input pin, and record the external input as z at this time (note that the external input is only connected from MSIO_0–MSIO_3).
4. Sample 1024 pieces of data and remove the maximum and minimum values to obtain the average of 1022 numbers.
5. At this time, the average value is recorded as x0, and the value of the P terminal voltage minus the N terminal voltage is recorded as y0.
6. Collect another voltage value using the above method, record the average code value as x1, and record the input voltage difference as y1.
7. Use the formula $y = (x - offset)/slope$, and bring in x0, y0, x1, y1 to obtain the offset and slope.
8. The raw data can be converted into the actual voltage using the following formula:

$$actual_voltage = (raw_data - offset)/(slope * z / vref) \quad (39)$$

where vref represents the voltage value of the external input as this time.

- Temperature sensor calibration

1. Select internal reference source by configuring the CFG.REF_SEL register, and select internal 0.8 V reference source by configuring the CFG.BAIS_RES_CTRL register.
2. Select single-ended mode by configuring the CFG.MODE register, and select temperature as N-terminal input by configuring the CFG.CH_N register to initialize ADC.
3. Put the chip into a constant temperature environment.
4. Sample 1024 pieces of data and remove the maximum and minimum values to obtain the average of 1022 numbers.
5. At this time, the average value is recorded as x0, and the current temperature is recorded as t0.
6. Put the chip into another constant temperature environment.
7. Sample 1024 pieces of data and remove the maximum and minimum values to obtain the average of 1022 numbers.
8. At this time, the average value is recorded as x1, and the current temperature is recorded as t1.
9. Use the formula $y = (x - offset)/slope$, and bring in x0, t0, x1, t1 to obtain the offset and slope.

10. The raw data can be converted into the actual temperature using the following formula:

$$actual_temperature = (raw_data - offset) / slope \quad (40)$$

- Battery voltage calibration

1. Select internal reference source by configuring the CFG.REF_SEL register, and select internal 0.8 V reference source by configuring the CFG.BAIS_RES_CTRL register.
2. Select single-ended mode by configuring the CFG.MODE register, and select battery as N-terminal input by configuring the CFG.CH_N register to initialize ADC.
3. Use a regulated power supply to supply power to the chip. Note that the power supply voltage of the chip that the ADC can work normally is 2.2 V–3.8 V.
4. Sample 1024 pieces of data and remove the maximum and minimum values to obtain the average of 1022 numbers.
5. At this time, the average value is recorded as x0, and the current battery voltage is recorded as y0.
6. Reselect another voltage value to power the chip.
7. Sample 1024 pieces of data and remove the maximum and minimum values to obtain the average of 1022 numbers.
8. At this time, the average value is recorded as x1, and the current battery voltage is recorded as y1.
9. Use the formula $y = (x - offset) / slope$, and bring in x0, y0, x1, y1 to obtain the offset and slope.
10. The raw data can be converted into the actual battery voltage using the following formula:

$$actual_bat_voltage = (raw_data - offset) / slope \quad (41)$$

8.8.1.10 Registers

8.8.1.10.1 CFG

- **Name:** Sense ADC Configuration Register
- **Description:** This register is used to configure reference, channels, and input mode of ADC.
- **Base Address:** 0x4000A800
- **Offset:** 0x00
- **Reset Value:** 0x0710070A

Table 8-604 Sense ADC Configuration Register

Bits	Field Name	RW	Reset	Description
31	RST	RW	0x0	Reset SADC. **Value:** • 0x0: None

Bits	Field Name	RW	Reset	Description
				• 0x1: Reset
30	EN	RW	0x0	Enable SADC. **Value:** • 0x0: Disable SADC • 0x1: Enable SADC
29:27	REF_SEL	RW	0x0	Reference voltage select **Value:** • 0x0: Buffered internal reference • 0x1: Reserved • 0x2: Reserved • 0x3: Using MSIO_0 as a reference • 0x4: Using MSIO_1 as a reference • 0x5: Using MSIO_2 as a reference • 0x6: Using MSIO_3 as a reference • 0x7: Reserved **Note:** Because the maximum input for any MSIO is supposed to be VBATT, there is no benefit from using an external input higher than VBATT/2.
26:24	REF_HP_MODE	RW	0x7	Used to define the current in the reference circuit: • 0x0: 100 ksps • 0x7: 1 Msps
23:20	CH_P	RW	0x1	Used to define input for channel P: **Value:** • 0x0–0x7: MSIO_0–MSIO_7 as input • 0xF: Internal reference as input
19:16	CH_N	RW	0x0	Used to define input for channel N: **Value:** • 0x0–0x7: MSIO_0–MSIO_7 as input • 0xD: Temperature sensor as input • 0xE: Battery sensor as input • 0xF: Internal reference as input
15	EN_TMP	RW	0x0	Enable temperature sensor. **Value:** • 0x0: Disable

Bits	Field Name	RW	Reset	Description
				• 0x1: Enable
14	EN_VBAT	RW	0x0	Enable battery sensor. **Value:** • 0x0: Disable • 0x1: Enable
13	MODE	RW	0x0	Used to define operation to either single ended or differential mode. **Value:** • 0x0: Differential • 0x1: Single ended
12	EN_CAL	RW	0x0	Used to swap inputs of comparator for offset calibration. **Value:** • 0x0: Disable • 0x1: Enable
11	RSVD	R		Reserved bit
10:8	CODE_IN	RW	0x7	Used to define dynamic range of ADC. Default: 0x7; for higher input signal frequencies close to Nyquist rate, use 0x1.
7:4	RSVD	R		Reserved bits
3:0	BIAS_RES_CTRL	RW	0xA	Buffered internal reference **Value:** • 0x3: 0.85 V • 0x7: 1.28 V • 0xA: 1.6 V • 0xF: 2.0 V **Note:** For any specific reference value (Vref), the maximum ADC input is supposed to be 2 x Vref.

8.8.1.10.2 FIFO_RD

- **Name:** Sense ADC FIFO Output Register
- **Description:** Reading this register will pop an entry from the sense ADC FIFO.
- **Base Address:** 0x4000E000
- **Offset:** 0x00
- **Reset Value:** 0x00000000

Table 8-605 Sense ADC FIFO Output Register

Bits	Field Name	RW	Reset	Description
31:0	DATA	R	0x0	When the register is read, data in the FIFO buffer is accessed.

8.8.1.10.3 FIFO_CTRL

- **Name:** Sense ADC FIFO Control Register
- **Description:** This register is used to enable DMA request and set the threshold at which the DMA request is made to the hardware handshake mechanism.
- **Base Address:** 0x4000E000
- **Offset:** 0x004
- **Reset Value:** 0x00010000

Table 8-606 Sense ADC FIFO Control Register

Bits	Field Name	RW	Reset	Description
31:17	RSVD	R		Reserved bits
16	DMA_EN	RW	0x1	• 1: DMA HW handshake from SNSADC FIFO is enabled • 0: No DMA HW handshake (suitable for polling mode)
15:6	RSVD	R		Reserved bits
5:0	THD	RW	0x0	This is used to set the threshold at which the DMA request is made to the hardware handshake mechanism.

8.8.1.10.4 FIFO_STAT

- **Name:** Sense ADC Status Register
- **Description:** This register contains the number of valid data entries in the ADC FIFO memory.
- **Base Address:** 0x4000E000
- **Offset:** 0x008
- **Reset Value:** 0x00000000

Table 8-607 Sense ADC Status Register

Bits	Field Name	RW	Reset	Description
31:17	RSVD	R		Reserved bits
16	FLUSH	W		Write 1 to flush FIFO DATA.
15:9	RSVD			Reserved bits
8	VAL	R	0x0	Identify whether FIFO is empty. **Value:**

Bits	Field Name	RW	Reset	Description
				• 0x0: FIFO is empty. • 0x1: FIFO is not empty.
7	RSVD	R		Reserved bit
6:0	COUNT	R	0x0	Sense ADC FIFO Count, number of 32-bit words in the FIFO. Maximum value is 0x40 or decimal 64.

8.8.1.10.5 CLK

- **Name:** Sense ADC Clock Register
- **Description:** This register selects ADC clock.
- **Base Address:** 0x4000E000
- **Offset:** 0x00C
- **Reset Value:** 0x00000000

Table 8-608 Sense ADC Clock Register

Bits	Field Name	RW	Reset	Description
31:19	RSVD	R		Reserved bits
18:16	CLK_SEL_RD	R		Actual clock select value
15:3	RSVD	R		Reserved bits
2:0	CLK_SEL	RW	0x0	SADC clock select **Value (Hz):** • 0: No clock • 1: 16k • 2: 8k • 3: 4k • 4: 16M • 5: 8M • 6: 4M • 7: 1M

8.8.1.10.6 ADC_GET_TKN_SW

- **Name:** Sense ADC Get Software Token Register
- **Description:** Read this register to try to get token for software and return the result of the trial.
- **Base Address:** 0x4000E000

- **Offset:** 0x018
- **Reset Value:** 0x00000000

Table 8-609 Sense ADC Get Software Token Register

Bits	Field Name	RW	Reset	Description
31:9	RSVD	R		Reserved bits
8	snsadc_tkn_owner_sw	R	0x0	• 0: Hardware has the token. • 1: Software has the token.
7:1	RSVD	R		Reserved bits
0	snsadc_tkn_locked_sw	R	0x0	• 0: Token available • 1: Token locked

8.8.1.10.7 ADC_RET_TKN_SW

- **Name:** Sense ADC Release Software Token Register
- **Description:** Write any value to this register to release the software token.
- **Base Address:** 0x4000E000
- **Offset:** 0x020
- **Reset Value:** 0x00000000

Table 8-610 Sense ADC Release Software Token Register

Bits	Field Name	RW	Reset	Description
31:1	RSVD	R		Reserved bits
0	snsadc_ret_tkn_sw	W	0x0	Write any value to this bit to release the SW token.

8.8.1.10.8 ADC_GET_TKN_STAT

- **Name:** Sense ADC Token Status Register
- **Description:** This register indicates the token status and the token owner.
- **Base Address:** 0x4000E000
- **Offset:** 0x024
- **Reset Value:** 0x00000000

Table 8-611 Sense ADC Token Status Register

Bits	Field Name	RW	Reset	Description
31:9	RSVD	R		Reserved bits
8	snsadc_tkn_owner	R	0x0	• 0: Hardware has the token. • 1: Software has the token.

Bits	Field Name	RW	Reset	Description
7:1	RSVD	R		Reserved bits
0	snsadc_tkn_locked	R	0x0	• 0: Token available • 1: Token locked

Note:

Reading this register would not issue any action of getting a token.

8.8.1.11 Electrical Specifications

Table 8-612 Electrical specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit
Physical number of bits	PNOB		13		bit
Sampling frequency	Freq	4 K		16 M	Hz
Integrated nonlinearity	INL	-8		8	LSB
Differential nonlinearity	DNL	-8		8	LSB
Signal-to-noise and distortion ratio	SNDR		60		dB
Total harmonic distortion	THD		65		dB
Effective number of bits	ENOB		9.5		bit
Differential offset voltage, 13-bit resolution	Vos	-4		4	LSB
Gain error	Eg	-1		1	%
Sampling capacitor	Cs		2		pF
Sampling rate	F_S	0.27		1067	ksps
Input resistance	Rin		> 1 M		ohm
Current consumption during ADC conversion @100 Ksps	I_ADC		< 20		µA
Spurious free dynamic range	SFDR		68		dB

8.8.2 Temperature Sensor

8.8.2.1 Introduction

A temperature sensor is integrated in GR5526 to measure the die temperature inside the chip. As one of the analog input signals of SADC, temperature signals can be converted into digital signals by using the SADC. You can obtain the temperature value according to the conversion formula. GR5526 also uses temperature sensors to adjust RF parameters inside the chip.

8.8.2.2 Main Features

- Integrated into the SADC module and serves as one of the SADC input signals.

- Measures the die temperature to check whether it is greater than or equal to the operating temperature of the device.
- Temperature measurement range: –40°C to + 80°C.
- The measurement accuracy at normal temperature is 1~2°C, and the maximum error at high or low temperature is 5°C.

8.8.2.3 Functional Description

The temperature sensor converts the temperature signals inside the chip into voltage. The SADC module inside the chip can be used to sample the converted voltage signals and then calculate the temperature value through the conversion formula.

8.8.2.3.1 Acquisition Parameter Configuration

In order to sample the voltage signals from the temperature sensor, SADC needs to be configured to single-ended mode: channel N should be set as ADC_INPUT_SRC_TMP, and enable temperature sensor by setting the bit of en_temp_sense. Set SADC as internal reference source, and select 0p8 for reference value. A 1 MHz clock is recommended for sampling clock.

8.8.2.3.2 Conversion Formula

After getting some code from the SADC, you can average the code and then convert the average code value into an actual temperature value according to the following conversion formula:

$$Temperature\ value = [(average_code - adc_temp)/(adc_slope * 2)]/(0.00175) + Temp_ref \quad (42)$$

In the formula above, adc_slope is the slope value corresponding to the reference voltage gear of SADC. The slope value corresponds to 0p8 gear. The adc_temp is the offset value after calibration based on normal temperature and the Temp_ref is the environment temperature value when calibrated.

These parameters are the calibration parameters of the chip and are saved in eFuse.

8.8.2.3.3 Measuring Internal Temperature

GR5526 has a temperature sensor module inside, and the measurement range is from -40°C to 80°C. The temperature sensor converts the internal temperature of the chip into a voltage value and outputs it to the ADC for sample and conversion. After temperature calibration, the ADC converts the digital value into an actual temperature value.

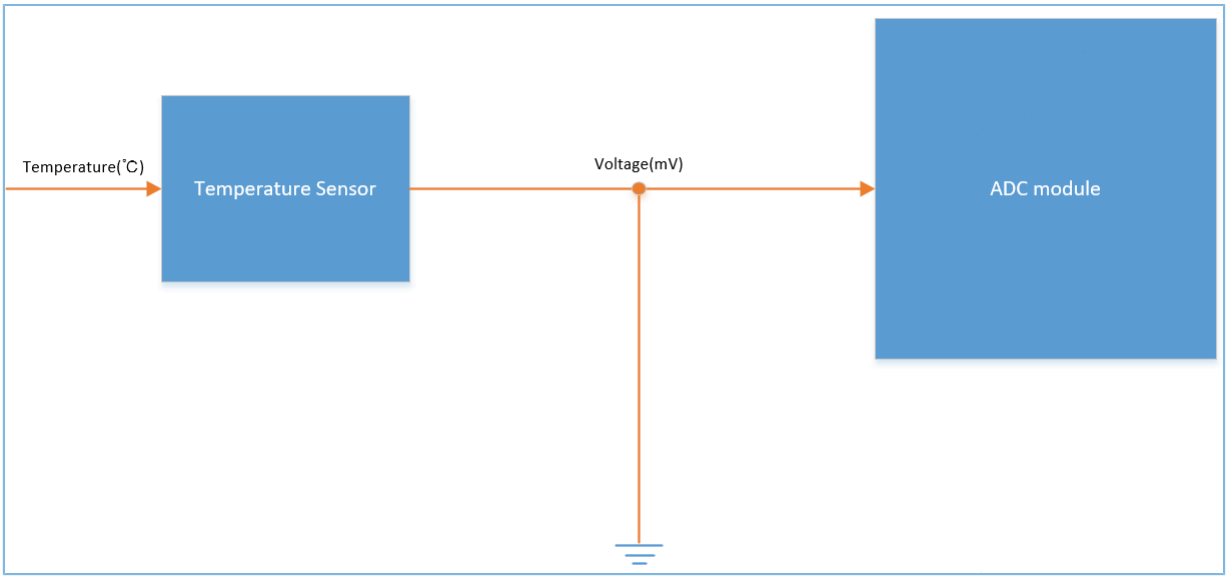

Figure 8-123 Schematic diagram of temperature measurement

If users want to measure the internal temperature of the chip, it is necessary to configure the N channel of the ADC as ADC_INPUT_SRC_TMP, and select the internal reference voltage of 0.8 V. The calibration formula is as follows:

$$temperature = (code - temp_25_offset) / slope / (-0.00175) + tem_25 \quad (43)$$

Where, temp_25 represents an ambient temperature of 25°C; temp_25_offset represents ADC raw value at 25°C ambient temperature; slope represents the calibrated slope value of the internal 0.8 V reference voltage; -0.00175 represents the amplification factor for the slope when used to measure temperature. In general, the output voltage decreases by a typical value of 1.863 mV for every 1 degree increase in temperature. The typical value of the voltage is 807.5 mV at -40°C and 583.9 mV at 80°C.

8.8.2.4 Registers

See the description of register CFG (Sense SADC Configuration Register) of SADC.

8.8.2.5 Electrical Specifications

Table 8-613 Electrical specifications

Parameter	Symbol	Min.	Typ.	Max.	Unit
Temperature sensor range	$Temp_{range}$	-40		80	°C
Temperature sensor accuracy	$Temp_{accuracy}$	-10		10	°C
Temperature sensor resolution	$Temp_{resolution}$		1.19		°C
Sample stability at constant temperature	$Temp_{stability}$	-1.19		1.19	°C
Voltage slope per degree	Vol_{degree}	-1.871	-1.863	-1.843	mV
Voltage of -40 degree	Vol_{-40}	807.5	807.5	809.9	mV
Voltage of 80 degree	Vol_{80}	581.5	581.5	583.9	mV

8.8.3 Comparator

8.8.3.1 Introduction

The comparator compares two analog signals: V1 (input voltage) and V2 (reference voltage), and generates a rising edge interrupt when V1 exceeds V2 or falling edge interrupt when V1 is lower than V2. Here it is assumed that V1 is connected to the positive side of the comparator, and V2 is connected to the negative side of the comparator. It is worth noting that interrupts will also be triggered when V1 is already lower than V2 before the comparator is enabled, or V2 is reset to a value higher than V1. The interrupt can be used to wake the system up from sleep state. These two signals can be derived from any of the eight MSIOs (MSIO_0–MSIO_7). Additionally, the comparator can be used to compare an external signal with one of the following:

- An internal programmable reference voltage.
- A programmable divided version of VBATL.

8.8.3.2 Main Features

- Input voltage (V1)
  - Internal reference voltage (Vref)
  - VBATL-based reference voltage (Vbatl_ref), internal reference based on VBATL
  - Eight MSIOs (MSIO_0–MSIO_7)
- Reference voltage (V2)
  - Internal reference voltage (Vref)
  - VBATL-based reference voltage (Vbatl_ref), internal reference based on VBATL
  - Eight MSIOs (MSIO_0–MSIO_7)
- Configurable 26 mV hysteresis
  - No hysteresis
  - –26 mV negative side of hysteresis
  - +26 mV positive side of hysteresis
  - Fully hysteresis 52 mV (±26 mV)
- Res degeneration is set for calibration.
  - Negative side calibration of comparator
  - Positive side calibration of comparator
- Bias current set (COMP_REG_0.ICOM_CTRL).
- Comparator channel selection

8.8.3.3 Functional Description

8.8.3.3.1 Overview

The following figure shows the overview of the wakeup comparator module.

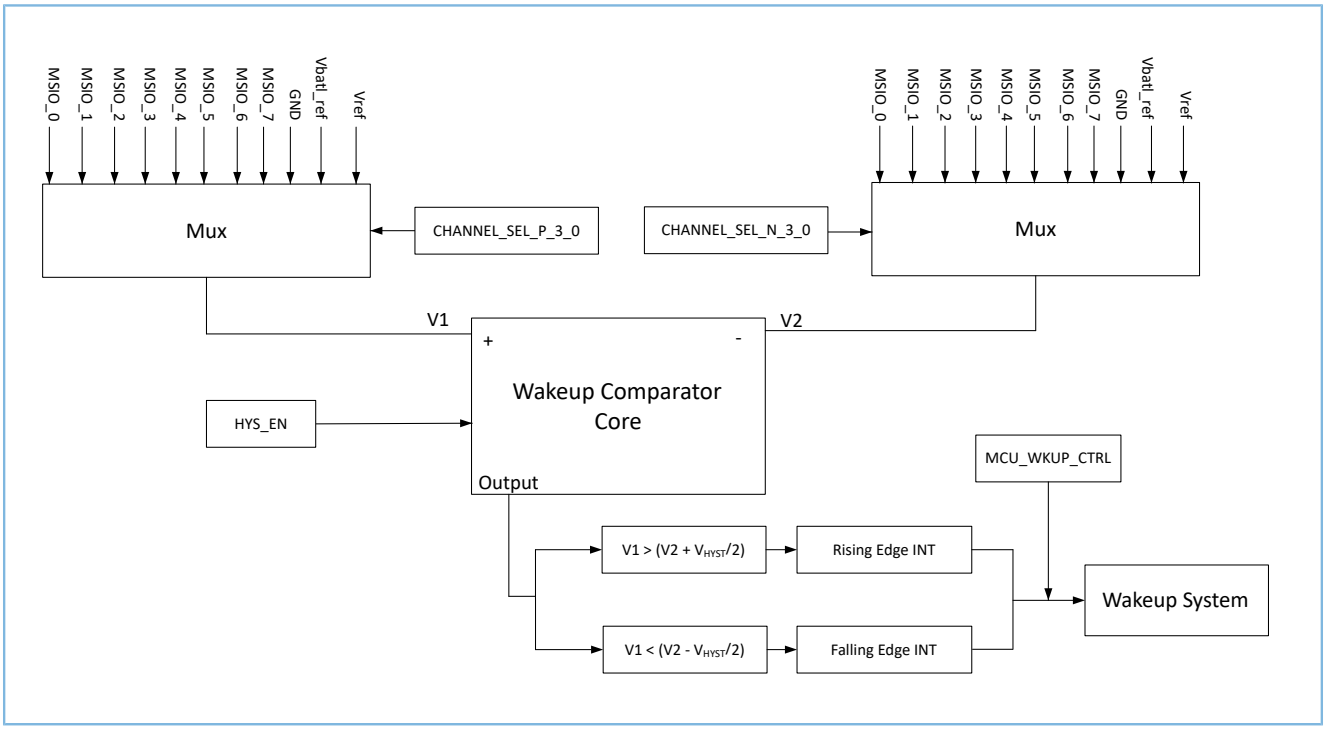

Figure 8-124 COMP overview

The Vbatl_ref comes from Vbatl through resistor divided, as shown in the Figure 8-125. Once enabled by using the COMP_REG_0.COMP_EN register, the comparator starts to work. When the wakeup comparator module is started, V1 starts to be compared with V2 continuously. When V1 exceeds V2, the wakeup interrupt will be triggered. The COMP_REG_0.CHANNEL_SEL_P_3_0 and COMP_REG_0.CHANNEL_SEL_N_3_0 registers are used to select the input of V1 and V2. When Vbatl_ref is selected in the COMP_REG_0.CHANNEL_SEL_N_3_0 or COMP_REG_0.CHANNEL_SEL_P_3_0 register, the bit COMP_REG_1.EN_VBATL_REF should be enabled first. The COMP_REG_0.COMP_VBATL_LVL_CTRL_2_0 register is used to select a level value of Vbatl_ref. The COMP_VBATL_LVL_CTRL_2_0 level value ranges from 0 to 7, and the calculation formula of Vbatl_ref is shown below:

$$Vbatl_ref = [(Level + 1)/10] \times VBATL \quad (44)$$

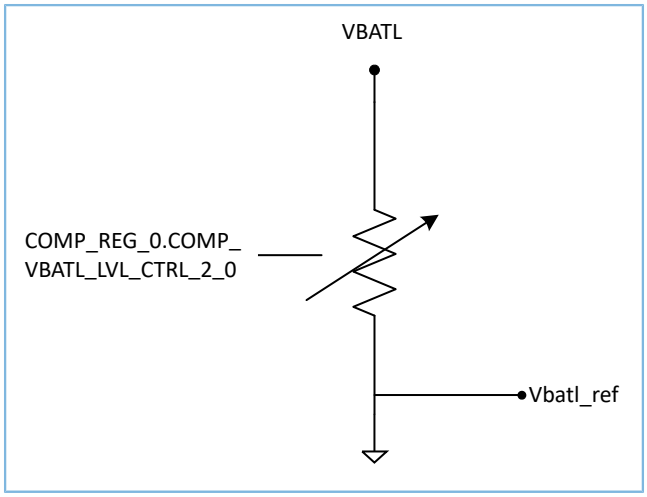

Figure 8-125 Set of Vbatl_ref control level

When Vref is selected in the COMP_REG_0.CHANNEL_SEL_P_3_0 or COMP_REG_0.CHANNEL_SEL_N_3_0 register, the bit COMP_REG_1.EN_REF should be enabled first. The COMP_REG_0.COMP_REF_CTRL_7_0 register is used to select a level value of Vref. The COMP_REF_CTRL_7_0 level value ranges from 0 to 255 and can be configured by COMP_REG_0.COMP_REF_CTRL_7_0 register. The calculation formula of Vref is shown below. Iref comes from the relatively constant current generated by the chip's bandgap module and will not be affected by other modules, as shown in the Figure 8-126.

$$Vref = 7.45 \quad mV \times Level \tag{45}$$

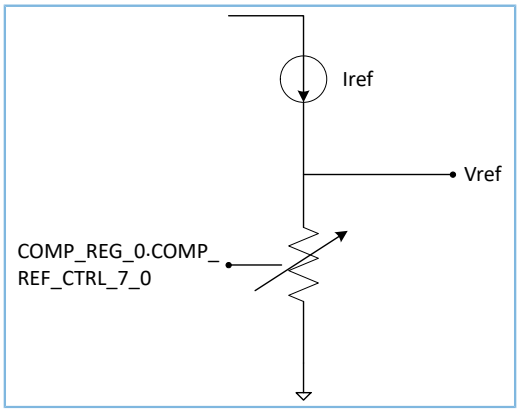

Figure 8-126 Set of Vref control level

Note:

When Vref is selected as reference, to keep the Iref stable, the chip supply voltage (VBATL) must be equal to or higher than 2.4 V.

The selected analog pins will be acquired by the comparator once it is enabled.

8.8.3.3.2 Hysteresis

The configurable 52 mV (±26 mV) hysteresis could be added into the wakeup comparator, which helps ensure the slowly changing voltage or a slightly noisy voltage that will cause the comparator not to output to oscillate frequently when the two input voltages are almost close. This could increase the anti-interference ability of the comparator. The hysteresis mode can be configured in COMP_REG_1.HYS_EN. The hysteresis on both sides mode is illustrated in Figure 8-127. The comparator will trigger a falling edge interrupt whenever V1 becomes lower than V2- (VHYST/2). It will also trigger a rising edge interrupt whenever V1 becomes higher than V2+ (VHYST/2). Here it is assumed that V1 is connected to the positive side of the comparator, and V2 is connected to the negative side of the comparator.

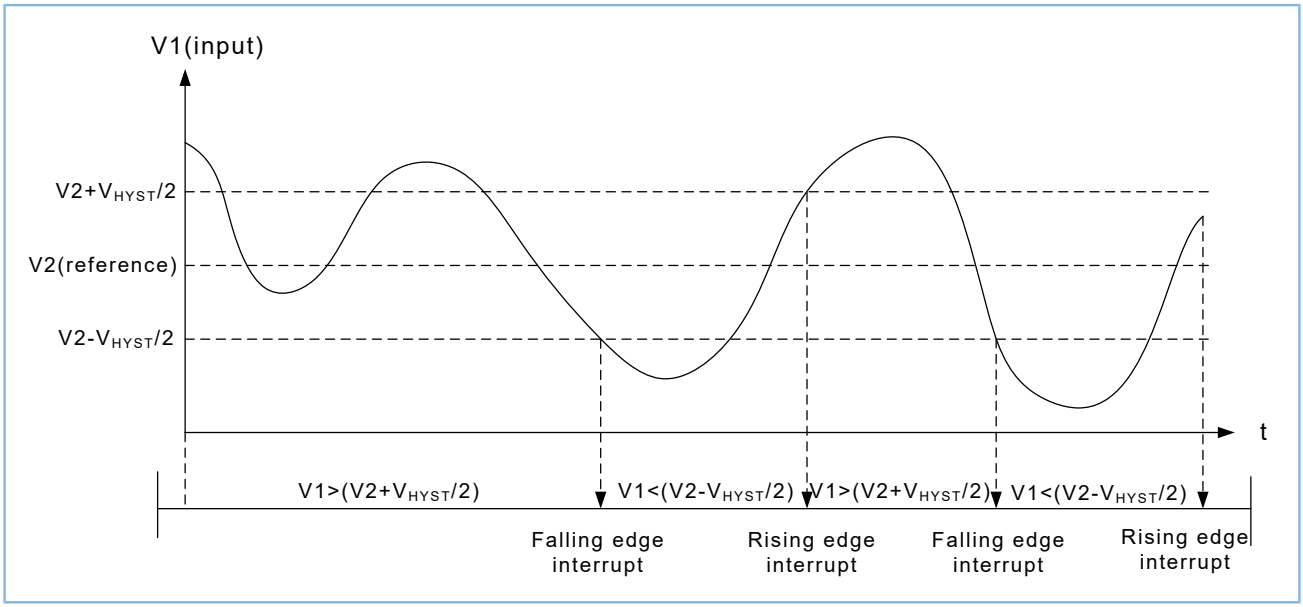

Figure 8-127 Both sides hysteresis enabled in wakeup comparator

8.8.3.3.3 Sample Clock

The clock source for collecting the comp output signal is shown in the figure below. The clock can be set as 250 kHz or 2 MHz by configuring the aon_slow_clk_sel. A higher clock frequency will improve the sampling accuracy, but correspondingly increase the power consumption.

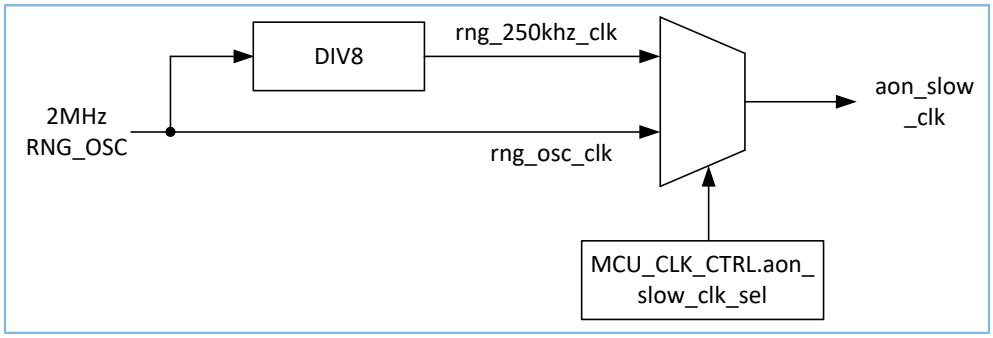

Figure 8-128 Comp output signal sample clock

8.8.3.3.4 Res Degeneration

Res degeneration is set for calibration. The default values of COMP_REG_1.RDEG_N_3_0 register and COMP_REG_1.RDEG_P_3_0 register are 0x8 and are always kept unchanged. The COMP module obtains positive

and negative res degeneration which means users can calibrate the comparator by adjusting positive and negative sides. For example, when the comparator outputs a valid signal, the voltage values at N and P sides of the comparator should be approximately equal; if it is found that the actual voltage at P side is too much larger than that at N side, you can set the value of COMP_REG_1.RDEG_P_3_0 smaller, or set the value of COMP_REG_1.RDEG_N_3_0 larger.

8.8.3.3.5 CascRes Half

The COMP_REG_1.CASCRES_HALF register is used to ensure that the comparator is in proper operation at different bias currents.

- The COMP_REG_1.CASCRES_HALF is set to 1 for half resistor when COMP_REG_0.ICOMP_CTRL_3_0 is close to 0000.
- The COMP_REG_1.CASCRES_HALF is set to 0 for full resistor when COMP_REG_0.ICOMP_CTRL_3_0 is close to 1111.

8.8.3.3.6 Bias Current Set

The COMP_REG_0. ICOMP_CTRL_3_0 register is used to configure the bias current. Higher bias currents will reduce the response time of the comparator, but with a corresponding increase in power consumption. Lower bias currents will increase the response time of the comparator, but with a corresponding reduce in power consumption. It is worth noting that this bias current is also from bandgap module, but this current and Iref are independent of each other and will not interfere with each other.

Glitch Remove

When the voltage reaches the threshold, the signal from the comp output will jitter because of noise interference. In order to eliminate jitter, users can specify a time by configuring the PMU_COMP_GLITCH_REMOVE.GLITCH_REMOVE_CYCLE register and the signal is considered valid if it does not change within this period of time. As the following figure shows, if the duration of the signal is less than the user set time Glitch_t, then the probability of triggering the interrupt will be greatly reduced. If the duration of the signal is larger than the user set time Glitch_t, then the interrupt will definitely be triggered.

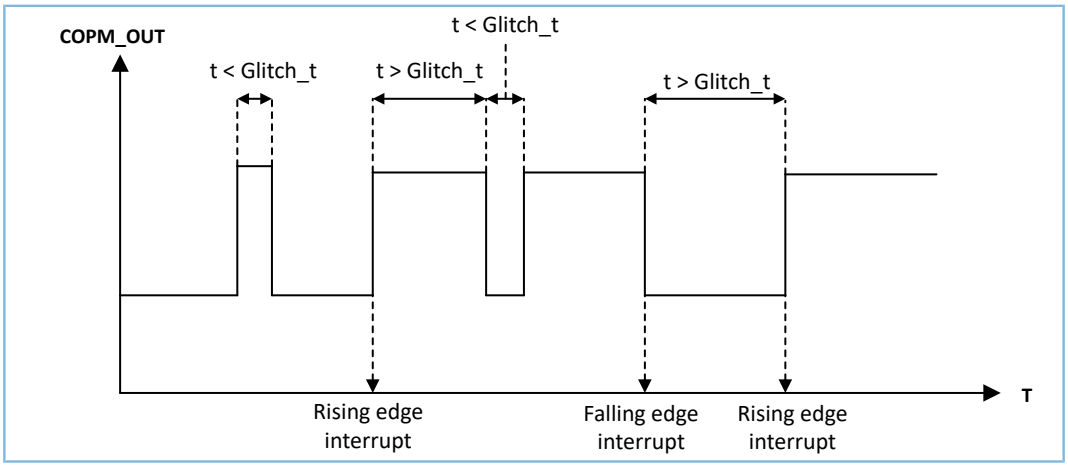

Figure 8-129 Glitch removing of comp

8.8.3.4 Registers

8.8.3.4.1 COMP_REG_0

- **Name:** Comparator Register 0
- **Description:** This register includes REF_VALUE setting, COMP enablement, and channel selection.
- **Base Address:** 0x4000A800
- **Offset:** 0x40
- **Reset Value:** 0x00000000

Table 8-614 Comparator Register 0

Bits	Field Name	RW	Reset	Description
23:20	CHANNEL_SEL_N_3_0	WR	0x0	Wakeup comparator channel N selection: • 0–7: MSIO_0–MSIO_7 • 8: GND • 9: Vbatl_ref • 10: Vref
19:16	CHANNEL_SEL_P_3_0	WR	0x0	Wakeup comparator channel P selection: • 0–7: MSIO_0–MSIO_7 • 8: GND • 9: Vbatl_ref • 10: Vref
15:8	COMP_REF_CTRL_7_0	WR	0x0	COMP REF CTRL (Vref) Changes the BG-based reference: Min = 0 V, Max =1.9 V (step = 7.5 mV). • 11111111: 1.9 V • 00000000: 0 V
7	COMP_EN	WR	0x0	Enables or disables wakeup comparator. • 0: Disable • 1: Enable
6:4	COMP_BATT_LVL_CTRL_2_0	WR	0x0	Changes the Vbatl based ref (Vbatl_vref): • 000: Vbatl/10 • 001: 2 Vbatl/10 … • 111 :8 Vbatl/10
3:0	ICOMP_CTRL_3_0	WR	0x0	• 0000: Max current (1 µA) • 1111: Min current (50 nA)

8.8.3.4.2 COMP_REG_1

- **Name:** Comparator Register 1
- **Description:** This register includes res degeneration, reference source enable, CascRes Half, and Hysteresis.
- **Base Address:** 0x4000A800
- **Offset:** 0x44
- **Reset Value:** 0x00000088

Table 8-615 Comparator Register 1

Bits	Field Name	RW	Reset	Description
15	EN_REF	WR	0x0	• 1: Enable internal Vref. • 0: Disable internal Vref.
14	EN_VBATL_REF	WR	0x0	• 1: Enable Vbatl_ref. • 0: Disable Vbatl_ref.
13	CASCRES_HALF	WR	0x0	CascRes Half: • 1: Half resistor (high power when ICOMP_CTRL_3_0 is close to 0000) • 0: Full resistor (low power when ICOMP_CTRL_3_0 is close to 1111)
12:10	RSVD	R	0x0	Reserved bits
9:8	HYS_EN	WR	0x0	• 00: No hysteresis • 01: Positive Hysteresis only • 10: Negative Hysteresis only • 11: Hysteresis on both sides Hysteresis = 26 mV
7:4	RDEG_N_3_0	WR	0x8	Res degeneration N, for calibration only
3:0	RDEG_P_3_0	WR	0x8	Res degeneration P, for calibration only

8.8.3.4.3 MCU_WKUP_CTRL

- **Name:** MCU Wakeup Control Register of COMP
- **Description:** For each bit, 0 indicates IRQ is disabled, and 1 indicates IRQ is enabled.
- **Base Address:** 0x4000A000
- **Offset:** 0x80
- **Reset Value:** 0x0000FFFF

Table 8-616 MCU Wakeup Control Register of COMP

Bits	Field Name	RW	Reset	Description
11	EVT_PMU_MSIO_COMP_F	WR	0X1	• 1: Enable the wakeup function of the comp falling edge.

Bits	Field Name	RW	Reset	Description
				• 0: Disable the wakeup function of the comp falling edge.
5	EVT_PMU_MSIO_COMP	WR	0x1	• 1: Enable the wakeup function of the comp rising edge. • 0: Disable the wakeup function of the comp rising edge.

8.8.3.4.4 AON_SLP_EVENT

- **Name:** AON Sleep Event Register of COMP
- **Description:** Events that happen in sleep or active mode, controlled by MCU_WKUP_CTRL.

Note:

When clearing the individual event, do not use read-mask-write but simply write **~ (1<< EVENT_BIT)** to AON_SLP_EVENT.

- **Base Address:** 0x4000A000
- **Offset:** 0x84
- **Reset Value:** 0x00000000

Table 8-617 AON Sleep Event Register of COMP

Bits	Field Name	RW	Reset	Description
11	EVT_PMU_MSIO_COMP_F	WR	0x0	Write 0 to this bit to clear comp falling event.
5	EVT_PMU_MSIO_COMP	WR	0x0	Write 0 to this bit to clear rising comp event.

8.8.3.4.5 PMU_COMP_GLITCH_REMOVE

- **Name:** COMP Glitch Remove Register
- **Description:** To remove voltage jitter, add deglitch circuit on comparator output (glitch: from 4 µs to 28 µs at a 4 µs step).
- **Base Address:** 0x4000A000
- **Offset:** 0x2C0
- **Reset Value:** 0x00000000

Table 8-618 COMP Glitch Remove Register

Bits	Field Name	RW	Reset	Description
31:3	RSVD	R		Reserved bits
2:0	GLITCH_REMOVE_CYCLE	WR	0	• 0: No glitch removal • 1–7: 4 µs*N glitch removal

8.8.3.5 Electrical Specifications

In the table below, the typical values are measured at 25 °C, and the minimum and maximum values are measured at the worst process and temperature conditions.

Table 8-619 Electrical specifications

Symbol	Description	Min.	Typ.	Max.	Unit
VREF	Internal reference voltage in wakeup comparator (when VBATL is equal to or higher than 3 V)	0		1900	mV
VBATL_REF	Battery based reference voltage in wakeup comparator	Vbatl*0.1		Vbatl*0.8	V
ERR_VREF	The error rate of the Internal reference voltage (when VBATL is equal to or higher than 3 V)		1.3	5	%
ERR_VBATL_REF	The error rate of the battery based reference voltage		1	5	%
P_HYS	Positive hysteresis voltage in wakeup comparator	16	26	36	mV
N_HYS	Negative hysteresis voltage in wakeup comparator	16	26	36	mV
T_RES	Comparator response time	8		12	µs

8.9 DMA

8.9.1 Introduction

The DMA module has two instances of Direct Memory Access (DMA) controller that performs fast data transfers from/to peripherals to/from on-chip SRAM over the on-chip bus. The DMA controller is highly programmable and allows the CPU to offload the burden of serving the slow-rate peripherals.

8.9.2 Main Features

- 2 DMA controller instances
- 6 DMA channels per DMA instance
- 16 handshaking interfaces per DMA instance
- Each DMA has 1 AHB master interface for moving data.
- Each DMA has 1 AHB slave interface for configuration.
- Maximum block size of 4 KB for each channel
- Scatter/Gather
- Block chaining
- Low power mode with global clock gating and channel clock gating

8.9.3 Block Diagram

Figure 8-130 shows the following functional groupings of the main interfaces to the DMA controller block:

- DMA hardware request interface

- 6 channels
- 128 bytes FIFO for Channel 0 and 32 bytes FIFO per other channel for source and destination
- Arbiter
- AHB master interface
- AHB slave interface

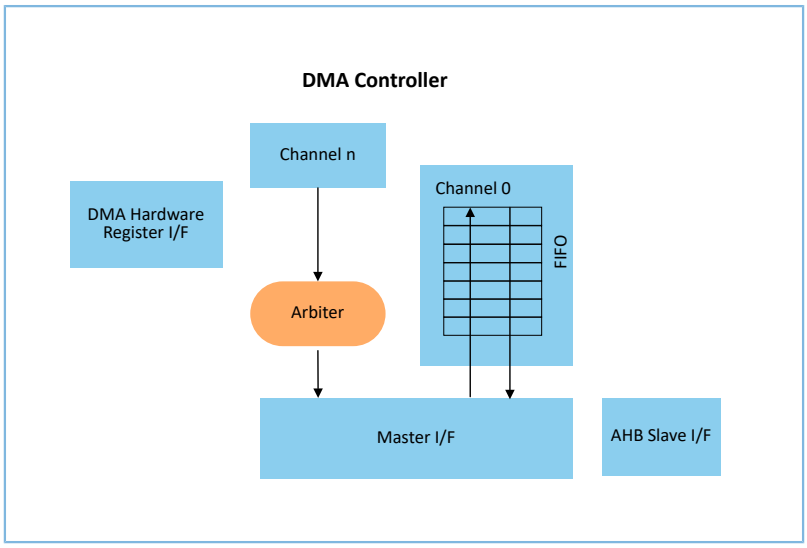

Figure 8-130 DMA controller block diagram

Figure 8-131 illustrates a peripheral-to-peripheral DMA transfer, where Peripheral A (source) uses a hardware handshaking interface, and Peripheral B (destination) uses a software handshaking interface. For example, the request to send data to Peripheral B is originated by the CPU, while writing to Peripheral B is handled by the DMA controller. The channel source and destination arbitrate independently for the AHB master interface, along with other channels.

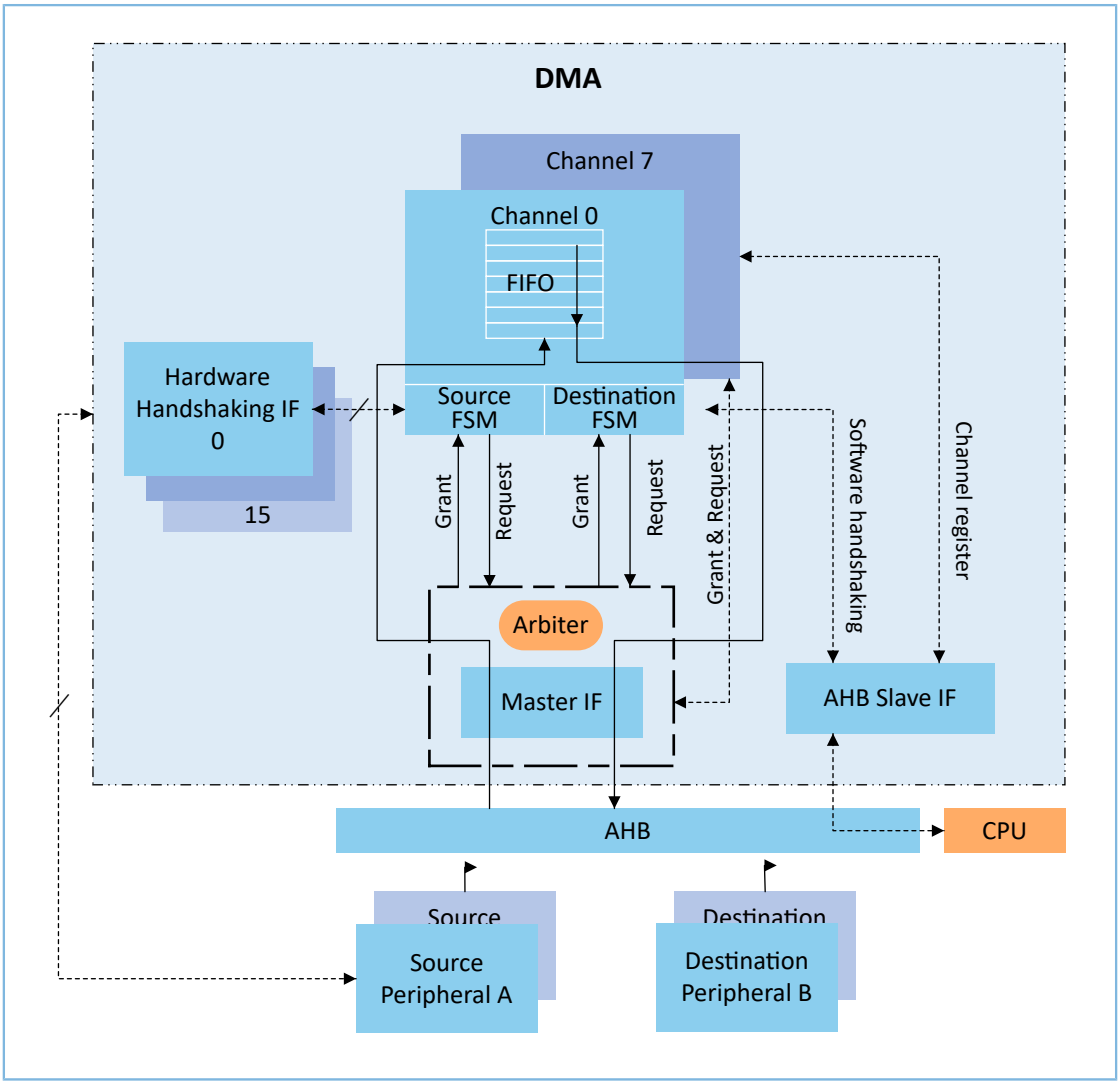

Figure 8-131 Peripheral-to-peripheral DMA transfer on the same AHB layer

8.9.4 Functional Description

8.9.4.1 Transfer Type Flow Control

GR5526 supports DMA as flow controller only. There are four transfer types:

- Memory to memory
- Memory to peripheral
- Peripheral to memory
- Peripheral to peripheral

8.9.4.2 Transfer Width

For memory, the source transfer width must be less than or equal to the data bus width of AHB master interface, normally 8, 16, or 32 bits. For a non-memory, the source transfer width is equal to the peripheral FIFO width.

For memory, the destination transfer width must be less than or equal to the data bus width of AHB master interface, normally 8, 16, or 32 bits. For a non-memory, the destination transfer width is equal to the peripheral FIFO width.

8.9.4.3 Source and Destination Address Increment

- Source address increment: It indicates whether to increase, decrease, or remain the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to No change.
- Destination address increment: This decides on whether to increment, decrement, or remain the destination address on every source transfer. If the device is fetching data from a destination peripheral FIFO with a fixed address, then set this field to **No change**.

8.9.4.4 Channel Priority

Each channel has a programmable priority in the range of 0–5 (each DMA has 6 channels), from lowest to highest. Channel priority applies to most scenarios except those on master bus. It is worth noting that a request for the master bus interface can be made at any time, but the request is granted only after the current AHB transfer (burst or single) completes. Therefore, when the master interface is transferring data for a low-priority channel while a high priority channel requests service, it completes the current AHB transfer (burst or single) for the low-priority channel before switching to data transfer for the high one.

8.9.4.5 DMA Transfers

DMA has 6 channels in total; each channel has a FIFO and is independent from each other. The FIFO of each channel is 32-bit wide. After the DMA channel is enabled, the DMA obtains data from the source according to the bit width configured for the source and stores the data into the DMA's FIFO. The DMA takes out the data from the DMA FIFO according to the bit width configured for the target and sends the data to the destination. Regardless of the configured transmission bit width, the data obtained from the source is compactly stored (this helps maximize the use of DAM's FIFO storage space). For example, the transmission bit width is 16 bits, and the FIFO bit width is 32 bits, so one depth of DMA FIFO will store two 16-bit data entries. It is worth mentioning that when DMA is used as flow control, the total number of bytes (blk_size_bytes_dma) transmitted by DMA in a single transmission depends on the product of the source transmission bit width (src_single_size_bytes) and the block size (CTLx.BLOCK_TS). The detailed calculation methods are as follows:

$$src_single_size_bytes = CTLx.SRC_TR_WIDTH/8 \tag{46}$$

$$blk_size_bytes_dma = CTLx.BLOCK_TS * src_single_size_bytes \tag{47}$$

Table 8-620 Transfer operations by source/destination bit width in DMA transfer

Source Interface Width	Destination Interface Width	Block Size (DMA as Flow Control)	Source Content: Address/ Data	Transfer Operations	Destination Content: Address/ Data
8	8	4	@0x0/B0 @0x1/B1	1: READ B0[7:0] @0x0 To DMA then WRITE B0[7:0] @0x0 2 :READ B1[7:0] @0x1 To DMA then WRITE B1[7:0] @0x1	@0x0/B0 @0x1/B1

Source Interface Width	Destination Interface Width	Block Size (DMA as Flow Control)	Source Content: Address/ Data	Transfer Operations	Destination Content: Address/ Data
			@0x2/B2 @0x3/B3	3: READ B2[7:0] @0x2 To DMA then WRITE B2[7:0] @0x2 4: READ B3[7:0] @0x3 To DMA then WRITE B3[7:0] @0x3	@0x2/B2 @0x3/B3
8	16	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1 :READ B0[7:0] @0x0 To DMA 2: READ B1[7:0] @0x1 To DMA then WRITE B1B0[15:0] @0x0 3: READ B2[7:0] @0x2 To DMA 4: READ B3[7:0] @0x3 To DMA then WRITE B3B2[15:0] @0x2	@0x0/B1B0 @0x2/B3B2
8	32	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1: READ B0[7:0] @0x0 To DMA 2: READ B1[7:0] @0x1 To DMA 3: READ B2[7:0] @0x2 To DMA 4: READ B3[7:0] @0x3 To DMA then WRITE B3B2B1B0[32:0] @0x0	@0x0/ B3B2B1B0
16	8	2	@0x0/B1B0 @0x2/B3B2	1: READ B1B0[15:0] @0x0 To DMA then WRITE B0[7:0] @0x0 2: WRITE B1[7:0] @0x1 3: READ B3B2[15:0] @0x2 To DMA then WRITE B2[7:0] @0x2 4: WRITE B3[7:0] @0x3	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3
16	16	2	@0x0/B1B0 @0x2/B3B2	1: READ B1B0[15:0] @0x0 To DMA then WRITE B1B0[15:0] @0x0 2: READ B3B2[15:0] @0x2 To DMA then WRITE B3B2[15:0] @0x2	@0x0/B1B0 @0x2/B3B2
16	32	2	@0x0/B1B0 @0x2/B3B2	1: READ B1B0[15:0] @0x0 To DMA 2: READ B3B2[15:0] @0x2 To DMA then WRITE B3B2B1B0[32:0] @0x0	@0x0/ B3B2B1B0
32	8	1	@0x0/ B3B2B1B0	1: READ B3B2B1B0[32:0] @0x0 To DMA then WRITE B0[7:0] @0x0 2: WRITE B1[7:0] @0x1 3: WRITE B2[7:0] @0x2 4: WRITE B3[7:0] @0x3	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3
32	16	1	@0x0/ B3B2B1B0	1: READ B3B2B1B0[32:0] @0x0 To DMA then WRITE B1B0[15:0] @0x0 2: WRITE B3B2[15:0] @0x2	@0x0/B1B0 @0x2/B3B2
32	32	1	@0x0/ B3B2B1B0	1: READ B3B2B1B0[32:0] @0x0 To DMA then WRITE B3B2B1B0[32:0] @0x0	@0x0/ B3B2B1B0

8.9.4.6 Gather and Scatter

Gather or scatter is a function that supports input and output data into/from multiple buffers. As shown in the figure below, gather refers to writing data fetched from multiple buffers to a channel (collecting data); scatter refers to reading data fetched from one channel to multiple buffers (scattering data).

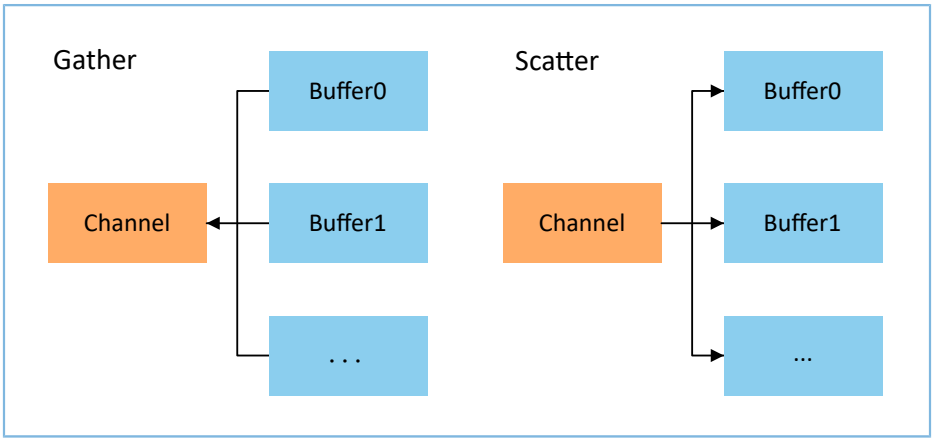

Figure 8-132 Gather/Scatter function

Gather is relevant to a source transfer. The source address is incremented or decremented by a programmed amount when a gather boundary is reached. The number of source transfers between successive gather boundaries is programmed into the Source Gather Count (SGRx.SGC) field. The source address is incremented or decremented by the value stored in the Source Gather Increment (SGRx.SGI) field multiplied by the number of bytes in a single AHB transfer from the source – (decoded value of CTLx.SRC_TR_WIDTH)/8 – when a gather boundary is reached. Gather is enabled by writing a 1 to the CTLx.SRC_GATHER_EN field.

Note:

If the CTLx.SINC field indicates a fixed-address control throughout a DMA transfer, then the CTLx.SRC_GATHER_EN field is ignored, and the gather feature is automatically disabled.

Figure 8-133 shows an example of a source gather transfer:

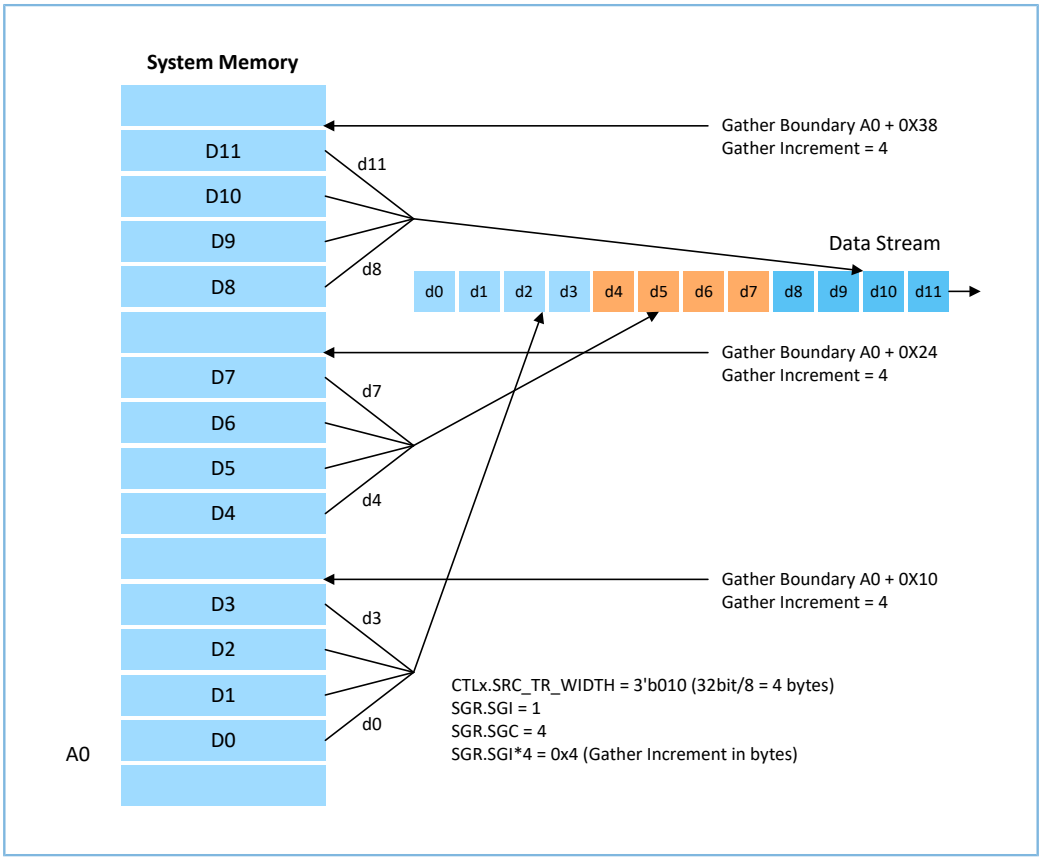

Figure 8-133 Source gather when SGR.SGI = 0x1

Scatter is relevant to a destination transfer. The destination address is incremented or decremented by a programmed amount – the scatter increment – when a scatter boundary is reached. The destination address is incremented or decremented by the value stored in the destination scatter increment (DSRx.DSI) field, multiplied by the number of bytes in a single AHB transfer to the destination (decoded value of CTLx.DST_TR_WIDTH)/8 – when a scatter boundary is reached. The number of destination transfers between successive scatter boundaries is programmed into the Destination Scatter Count (DSC) field of the DSRx register. Scatter is enabled by writing a 1 to the CTLx.DST_SCATTER_EN field.

Note:

If the CTLx.DINC field indicates a fixed-address control throughout a DMA transfer, then the CTLx.DST_SCATTER_EN field is ignored, and the scatter feature is automatically disabled.

Figure 8-134 shows an example of a destination scatter transfer:

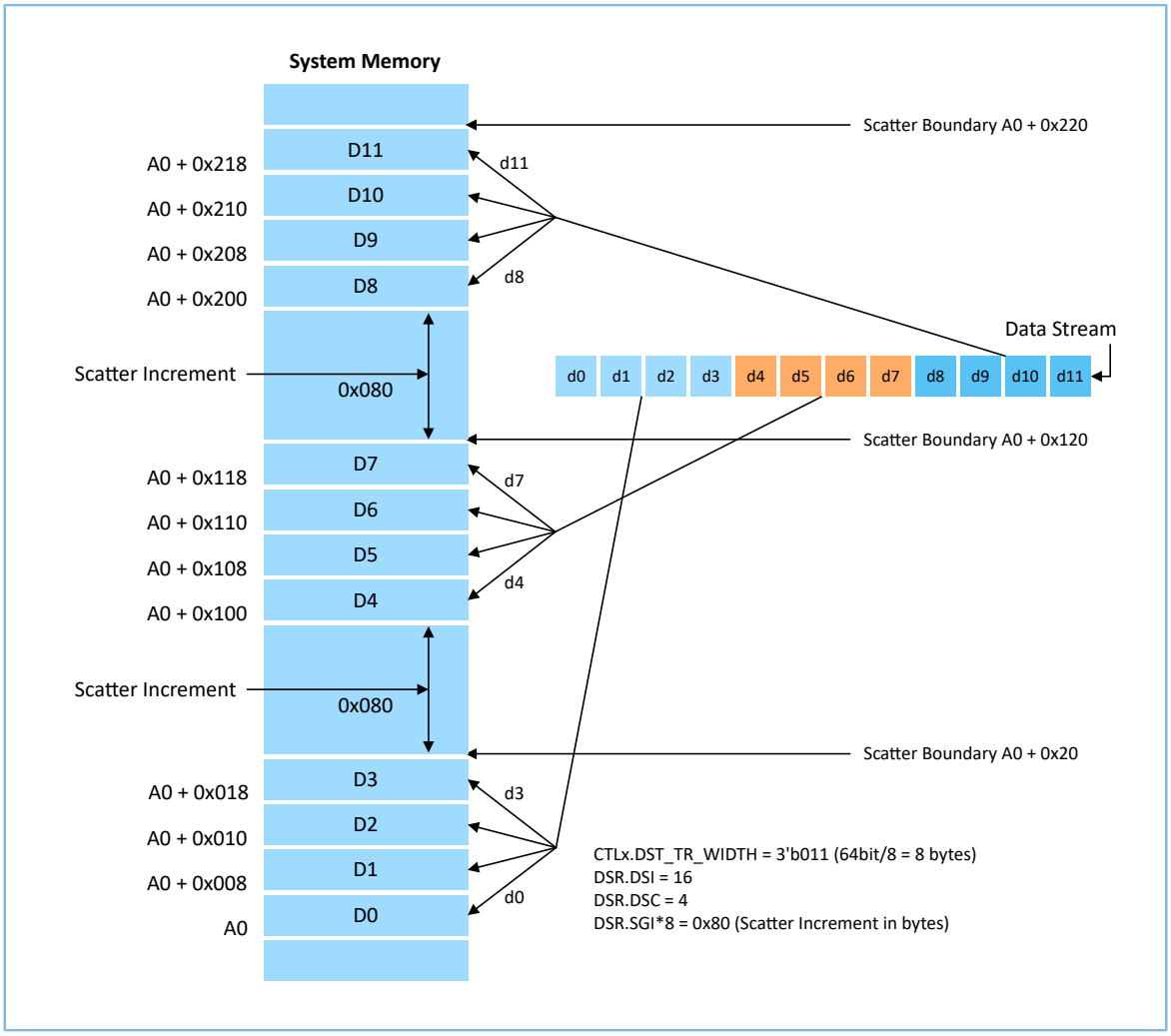

Figure 8-134 Example of destination scatter transfer

8.9.4.7 Multi-block DMA Transfer

DMA transfer may consist of multiple DW_AHB_DMAC blocks. Multi-block DMA transfers are supported through block chaining (linked list pointers) and auto-reloading channel registers. The DMA transfer modes for source and destination can be independently selected.

8.9.4.7.1 Auto-reloading

During auto-reloading, the channel registers are reloaded with their initial values at the completion of each block and the new values used for the new block. Some or all of the SRC_ADDR_CHx, DEST_ADDR_CHx, and CTRL_CHx channel registers are reloaded from their initial values at the start of a block transfer.

By configuring the Configuration Register (CFG_CHx.RELOAD_SRC), you can program the register if the SRC_ADDR_CHx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers.

By configuring the Configuration Register (CFG_CHx.RELOAD_DST), you can program the register if the DEST_ADDR_CHx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers.

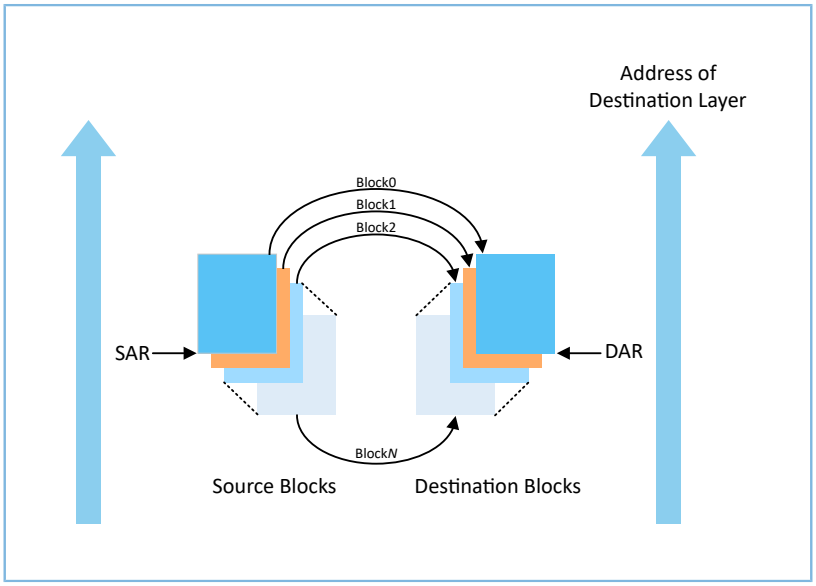

Figure 8-135 Multi-block DMA transfer with source and destination address auto-reloaded

8.9.4.7.2 Linked Lists (Block Chaining)

Linked list pointer (LLP) points to the location in system memory where the next linked list item (LLI) exists. The LLI is a set of registers that describe the next block (block descriptor) and an LLP register. The DMA fetches the LLI at the beginning of every block when block chaining is enabled. Each LLI is composed of the following parts: SAR, DAR, LLP, CTL, SSTAT, and DSTAT. Therefore, to use block chaining, a series of linked lists need to be implemented in memory. The LLP bit field of each LLI block stores the first address of the next LLI, thus forming a linked list of blocks. The following figure shows how to use chained linked lists in memory to define multi-block transfers using block chaining.

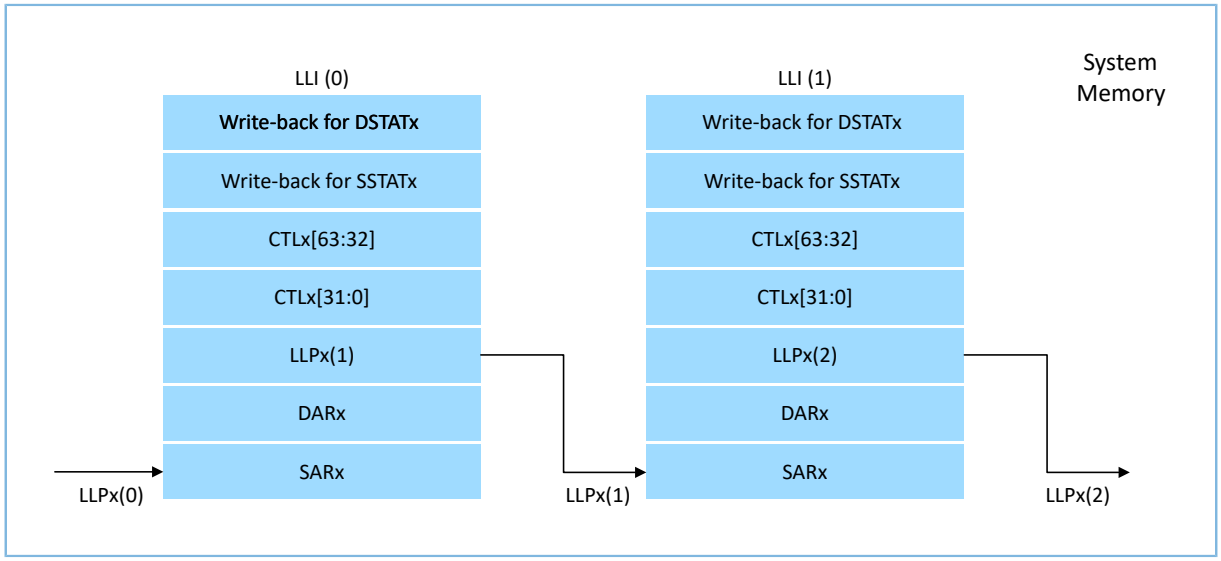

Figure 8-136 Multi-block transfer using linked lists when DMAH_CHx_STAT_SRC set to true

The DMA reprograms the channel registers prior to the start of each block by fetching the block descriptor for that block from system memory. This is known as an LLI update. The SARx, DARx, LLPx, and CTLx registers are fetched

from system memory on an LLI update. After that, the updated contents of the CTLx, SSTATx, and DSTATx registers are written back to memory on block completion.

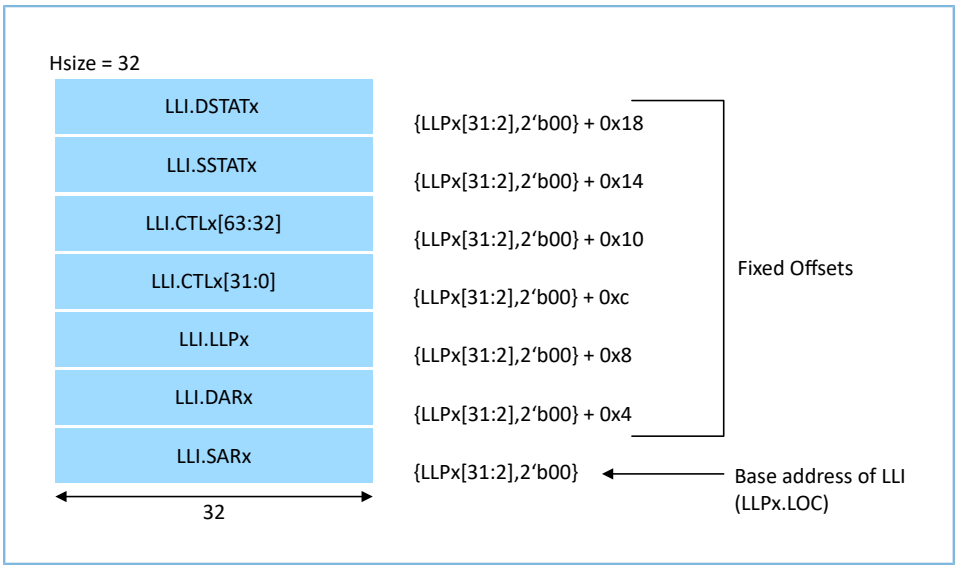

Figure 8-137 Mapping of block descriptor (LLI) in memory to channel registers when DMAH_CHx_STAT_SRC set to true

Note:

LLI accesses are always 32-bit accesses (**Hsize** = **2**) aligned to 32-bit boundaries and cannot be changed or programmed to anything other than 32-bit, even if the AHB master interface of the LLI supports more than a 32-bit data width.

8.9.4.8 DMA Interrupt

For each channel, there are three types of interrupt sources:

- IntTfr: DMA Transfer Complete Interrupt
- IntBlock: Block Transfer Complete Interrupt. This interrupt is generated on DMA block transfer completion to the destination peripheral.
- IntErr: Error Interrupt. This interrupt is generated when an ERROR response is received from an AHB slave on the HRESP bus during a DMA transfer. In addition, the DMA transfer is cancelled, and the channel is disabled.

The INT_RSTAT, INT_STAT, INT_MASK, and INT_CLR registers are provided to manage interrupts. The INT_MASK registers is used to enable or disable the block transfer complete interrupt, DMA transfer complete interrupt, and error interrupt. By reading the INT_STAT registers or INT_RSTAT registers, you can check whether the interruption have occurred. To clear interrupts, write 1 to the INT_CLR registers.

- INT_MASK registers: Used to enable or disable the block transfer complete interrupt, DMA transfer complete interrupt, and error interrupt.
- INT_STAT or INT_RSTAT registers: Used to reply whether the interruption have occurred.
- INT_CLR registers: Write 1 to INT_CLR to clear interrupts.

8.9.4.9 DMA Hardware Interface Assignment

Each channel of DMA can communicate with designated peripheral devices through the assigned hardware handshake interface. The hardware handshaking interfaces are assigned as shown in the following table. The developer can program the CFG_CHx.DEST_PER or CFG_CHx.SRC_PER to assign a handshaking interface for each channel source and destination.

Table 8-621 DMA0 hardware handshaking interfaces

DMA0 Request	Hardware Handshake Choice0	Handshake Peripheral Description	Hardware Handshake Choice1	Handshake Peripheral Description
0x0	QSPI0 TX	QSPI0 Master Transmit	NC	/
0x1	QSPI0 RX	QSPI0 Master Receive	NC	/
0x2	SPIM TX	SPI Master Transmit	I2C2 TX	I2C2 Transmit
0x3	SPIM RX	SPI Master Receive	I2C2 RX	I2C2 Receive
0x4	SPIS TX	SPI Slave Transmit	UART3 TX	UART3 Transmit
0x5	SPIS RX	SPI Slave Receive	UART3 RX	UART3 Receive
0x6	UART0 TX	UART0 Transmit	I2C5 TX	I2C5 Transmit
0x7	UART0 RX	UART0 Receive	I2C5 RX	I2C5 Receive
0x8	UART1 TX	UART1 Transmit	I2C4 TX	I2C4 Transmit
0x9	UART1 RX	UART1 Receive	I2C4 RX	I2C4 Receive
0xA	SNSADC	Sense ADC	UART4 TX	UART4 Transmit
0xB	NC	/	UART4 RX	UART4 Receive
0xC	OSPI TX	OSPI Master Transmit	QSPI1 TX	QSPI1 Master Transmit
0xD	OSPI RX	OSPI Master Receive	QSPI1 RX	QSPI1 Master Receive
0xE	UART2 TX	UART2 Transmit	I2C3 TX	I2C3 Transmit
0xF	UART2 RX	UART2 Receive	I2C3 RX	I2C3 Receive

Table 8-622 DMA1 hardware handshaking interfaces

DMA1 Request	Hardware Handshake Choice0	Handshake Peripheral Description	Hardware Handshake Choice1	Handshake Peripheral Description
0x0	OSPI TX	OSPI Master Transmit	SPIM TX	SPI Master Transmit
0x1	OSPI RX	OSPI Master Receive	SPIM RX	SPI Master Receive
0x2	QSPI2 TX	QSPI2 Master Transmit	DSPIM TX	DSPI Master Transmit
0x3	QSPI2 RX	QSPI2 Master Receive	DSPIM RX	DSPI Master Receive
0x4	I2SM TX	I2S Master Transmit	QSPI1 TX	QSPI1 Master Transmit
0x5	I2SM RX	I2S Master Receive	QSPI1 RX	QSPI1 Master Receive
0x6	I2SS TX	I2S Slave Transmit	UART3 TX	UART3 Transmit

DMA1 Request	Hardware Handshake Choice0	Handshake Peripheral Description	Hardware Handshake Choice1	Handshake Peripheral Description
0x7	I2SS RX	I2S Slave Receive	UART3 RX	UART3 Receive
0x8	PDM	PDM Receive	UART4 TX	UART4 Transmit
0x9	NC	/	UART4 RX	UART4 Receive
0xA	QSPI1 TX	QSPI1 Master Transmit	UART5 TX	UART5 Transmit
0xB	QSPI1 RX	QSPI1 Master Receive	UART5 RX	UART5 Receive
0xC	I2C0 TX	I2C0 Transmit	NC	/
0xD	I2C0 RX	I2C0 Receive	NC	/
0xE	I2C1 TX	I2C1 Transmit	UART0 TX	UART0 Transmit
0xF	I2C1 RX	I2C1 Receive	UART0 RX	UART0 Receive

8.9.5 Registers

8.9.5.1 Channel_x_Registers

8.9.5.1.1 SARx

- **Name**: Source Address for Channel x
- **Description**: The starting source address is programmed by software before the DMA channel is enabled, or by an LLI update before the start of the DMA transfer. While the DMA transfer is in progress, this register is updated to reflect the source address of the current AHB transfer.

Note:

You must program the SAR address to be aligned to CTLx.SRC_TR_WIDTH.

- **Base Address**: 0x40013000
- **Offset**: 0x0 + x*0x58
- **Reset Value**: 0x0

Table 8-623 Source Address Register for Channel x

Bits	Field Name	RW	Reset	Description
63:32	RSVD	R		Reserved field
31:0	SRC_ADDR	RW	0x0	Current Source Address of DMA transfer. Updated after each source transfer. The SRC_ADDR_INC field in the CTRL_CHx register determines whether the address increments, decrements, or is left unchanged on every source transfer through the block transfer.

Bits	Field Name	RW	Reset	Description
				Volatile: true

8.9.5.1.2 DARx

- **Name**: Destination Address Register for Channel x
- **Description**: The starting destination address is programmed by software before the DMA channel is enabled, or by an LLI update before the start of the DMA transfer. While the DMA transfer is in progress, this register is updated to reflect the destination address of the current AHB transfer.

Note:

You must program the DAR to be aligned to CTLx.DST_TR_WIDTH.

- **Base Address**: 0x40013000
- **Offset**: 0x8 + x*0x58
- **Reset Value**: 0x0

Table 8-624 Destination Address Register for Channel x

Bits	Field Name	RW	Reset	Description
63:32	RSVD	R		Reserved field
31:0	DEST_ADDR	RW	0x0	Current Destination address of DMA transfer. Updated after each destination transfer. The DEST_ADDR_INC field in the CTRL_CHx register determines whether the address increments, decrements, or is left unchanged on every destination transfer throughout the block transfer. Volatile: true

8.9.5.1.3 LLPx

- **Name**: Linked List Pointer Register for Channel x
- **Description**: The LLP register has two functions:
  - The logical result of the equation LLP.LOC !=0 is used to set up the type of DMA transfer: single or multi-block. If LLP.LOC is set to 0x0, transfers using linked lists are not enabled. This register must be programmed prior to enabling the channel in order to set up the transfer type.
  - LLP.LOC != 0 contains the pointer to the next LLI for block chaining using linked lists. The LLPx register can also point to the address where write back of the control and source/destination status information occurs after block completion.
- **Base Address**: 0x40013000

- **Offset**: 0x10 + x*0x58
- **Reset Value**: 0x0

Table 8-625 Linked List Pointer Register for Channel x

Bits	Field Name	RW	Reset	Description
63:32	RSVD	R		Reserved field
31:2	LOC	RW	0x0	Starting Address In Memory of next LLI if block chaining is enabled. Note that the two LSBs of the starting address are not stored because the address is assumed to be aligned to a 32-bit boundary. LLI accesses are always 32-bit accesses (Hsize = 2) aligned to 32-bit boundaries and cannot be changed or programmed to anything other than 32-bit. Volatile: true
1:0	RSVD	R		Reserved field

8.9.5.1.4 CTLx

- **Name**: Control Register for Channel x
- **Description**: This register contains fields that control the DMA transfer. The CTLx register is part of the block descriptor (linked list item - LLI) when block chaining is enabled. It can be varied on a block-by-block basis within a DMA transfer when block chaining is enabled. If status write-back is enabled, the upper word of the control register, CTLx[63:32], is written to the control register location of the LLI in system memory at the end of the block transfer.

Note:

You need to program this register prior to enabling the channel.

- **Base Address**: 0x40013000
- **Offset**: 0x18 + x*0x58
- **Reset Value**: 0x0000000200304801

Table 8-626 Control Register for Channel x

Bits	Field Name	RW	Reset	Description
63:45	RSVD	R	0x0	Reserved field Volatile: true
44	DONE	RW	0x0	Done bit. If status write-back is enabled, the upper word of the control register, CTRL_CHx[63:32], is written to the control

Bits	Field Name	RW	Reset	Description
				register location of the LLI in system memory at the end of the block transfer with the done bit set. Software can poll the LLI CTRL_CHx.DONE bit to see when a block transfer completes. The LLI CTRL_CHx.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel. LLI accesses are always 32-bit accesses (Hsize = 2) aligned to 32-bit boundaries and cannot be changed or programmed to anything other than 32-bit. Volatile: true
43:32	BLOCK_XFE_SIZE	RW	0x2	Block Transfer Size. When the DMA is the flow controller, users write this field before the channel is enabled in order to indicate the block size. The number programmed into BLOCK_XFE_SIZE indicates the total number of single transactions to perform for every block transfer; a single transaction is mapped to a single AMBA beat. Width: The width of single transaction is determined by CTRL_CHx.SRC_XFE_WIDTH. Once the transfer starts, the read-back value is the total number of data items already read from the source peripheral, regardless of what the flow controller is. When the source or destination peripheral is assigned as the flow controller, the maximum block size that can be read back saturates at 0xFFF, but the actual block size can be greater. Volatile: true
31:29	Rsvd_1_CTRL	R	0x0	Reserved field Volatile: true
28	LLP_SRC_EN	RW	0x0	Block chaining is enabled on the source side only if the LLP_SRC_EN field is high and LLPx.LOC is non-zero. **Value**: • 0x0 (LLP_SRC_DISABLE): Block chaining using linked list is disabled on the source side. • 0x1 (LLP_SRC_ENABLE): Block chaining using linked list is enabled on the source side.

Bits	Field Name	RW	Reset	Description
				• 0x0 (DISABLED): DONE bit is deasserted at the end of block transfer. • 0x1 (ENABLED): SET the DONE bit at the end of block transfer. Volatile: true
27	LLP_DST_EN	RW	0x0	Block chaining is enabled on the destination side only if LLP_DST_EN field is high and LLPx.LOC is non-zero. **Value**: • 0x0 (LLP_DST_DISABLE): Block chaining using linked list is disabled on the destination side. • 0x1 (LLP_DST_ENABLE): Block chaining using linked list is enabled on the destination side. Volatile: true
26:23	Rsvd	R	0x0	Reserved field Volatile: true
22:20	XFE_TYPE_FC	RW	0x3	Transfer Type and Flow Control. Flow control can be assigned to the DMA, the source peripheral, or the destination peripheral. **Value**: • 0x0: Transfer type is memory to memory, and flow controller is DMA. • 0x1: Transfer type is memory to peripheral, and flow controller is DMA. • 0x2: Transfer type is Peripheral to memory, and flow controller is DMA. • 0x3: Transfer type is peripheral to peripheral, and flow controller is DMA. Volatile: true
19	Rsvd_CTRL	R	0x0	Reserved field Volatile: true
18	DST_SCATTER_EN	RW	0x0	Destination scatter enable. Scatter on the destination side is applicable only when the CTLx.DINC bit indicates an incrementing or decrementing address control. **Value**:

Bits	Field Name	RW	Reset	Description
				• 0x0 (DST_SCATTER_DISABLE): Destination scatter is disabled. • 0x1 (DST_SCATTER_ENABLE): Destination scatter is enabled. Volatile: true
17	SRC_GATHER_EN	RW	0x0	Source gather enable. Gather on the source side is applicable only when the CTLx.DINC bit indicates an incrementing or decrementing address control. **Value**: • 0x0 (SRC_GATHER_DISABLE): Source gather is disabled. • 0x1 (SRC_GATHER_ENABLE): Source gather is enabled. Volatile: true
16:14	SRC_MSIZE	RW	0x1	Source burst transaction length. Number of data items, each of width CTRL_CHx.SRC_XFE_WIDTH, to be read from the source every time a burst transferred request is made from either the corresponding hardware or software handshaking interface. **Note**: This value is not related to the AHB bus master HBURST bus. **Value**: • 0x0: Number of data items to be transferred is 1. • 0x1: Number of data items to be transferred is 4. • 0x2: Number of data items to be transferred is 8. Volatile: true
13:11	DEST_MSIZE	RW	0x1	Destination burst transaction length. Number of data items, each of width CTRL_CHx.DEST_XFE_WIDTH, to be written to the destination every time a destination burst transaction request is made from either the corresponding hardware or software handshaking interface. **Note**: This value is not related to the AHB bus master HBURST bus. **Value**:

Bits	Field Name	RW	Reset	Description
				• 0x0: Number of data items to be transferred is 1. • 0x1: Number of data items to be transferred is 4. • 0x2: Number of data items to be transferred is 8. Volatile: true
10:9	SRC_ADDR_INC	RW	0x0	Source address increment. Indicates whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, set this field to "No change". **Value**: • 0x0: Increments the source address. • 0x1: Decrements the source address. • 0x2/0x3: No change in the source address Volatile: true
8:7	DEST_ADDR_INC	RW	0x0	Destination address increment. Indicates whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, set this field to "No Change". **Value**: • 0x0: Increments the destination address. • 0x1: Decrements the destination address. • 0x2/0x3: No change in the destination address Volatile: true
6:4	SRC_XFE_WIDTH	RW	0x0	Source Transfer Width. Mapped to AHB bus Hsize. For a non-memory peripheral, typically the peripheral (source) FIFO width. **Value**: • 0x0: Source transfer width is 8 bits. • 0x1: Source transfer width is 16 bits. • 0x2: Source transfer width is 32 bits. Volatile: true
3:1	DEST_XFE_WIDTH	RW	0x0	Destination Transfer Width.

Bits	Field Name	RW	Reset	Description
				Mapped to AHB bus Hsize. For a non-memory peripheral, typically the peripheral (destination) FIFO width. **Value**: • 0x0: Destination transfer width is 8 bits. • 0x1: Destination transfer width is 16 bits. • 0x2: Destination transfer width is 32 bits. Volatile: true
0	INT_EN	RW	0x1	Interrupt Enable Bit. If set, all interrupt-generating sources are enabled. Functions as a global mask bit for all interrupts for the channel; raw interrupt registers still assert if CTRL_CHx.INT_EN = 0. **Value**: • 0x0: Interrupt is disabled. • 0x1: Interrupt is enabled. Volatile: true

8.9.5.1.5 SSTATx

- **Name**: Source Status Register for Channel x
- **Description**: After each block transfer completes, hardware can retrieve the source status information from the address pointed to by the contents of the SSTATARx register. This status information is then stored in the SSTATx register and written out to the SSTATx register location of the LLI before the start of the next block. For conditions under which the source status information is fetched, it can be varied on a block-by-block basis within a DMA transfer when block chaining is enabled. If status write-back is enabled, the upper word of the control register, CTLx[63:32], is written to the control register location of the LLI in system memory at the end of the block transfer.

Note:

This register is a temporary placeholder for the source status information on its way to the SSTATx register location of the LLI. The source status information should be retrieved by software from the SSTATx register location of the LLI, and not by a read of this register over the DMA controller slave interface.

- **Base Address**: 0x40013000
- **Offset**: 0x20 + x*0x58
- **Reset Value**: 0x0

Table 8-627 Source Status Register for Channel x

Bits	Field Name	RW	Reset	Description
63:32	RSVD	R		Reserved field
31:0	SSTAT	RW	0x0	Source status information retrieved by hardware from the address pointed to by the contents of the STATARx register Volatile: true

8.9.5.1.6 DSTATx

- **Name**: Destination Status Register for Channel x
- **Description**: After each block transfer completes, hardware can retrieve the destination status information from the address pointed to by the contents of the DSTATARx register. This status information is then stored in the DSTATx register and written out to the DSTATx register location of the LLI before the start of the next block.

Note:

This register is a temporary placeholder for the destination status information on its way to the DSTATx register location of the LLI. The destination status information should be retrieved by software from the DSTATx register location of the LLI, and not by a read of this register over the DMAC controller slave interface.

- **Base Address**: 0x40013000
- **Offset**: 0x28 + x*0x58
- **Reset Value**: 0x0

Table 8-628 Destination Status Register for Channel x

Bits	Field Name	RW	Reset	Description
63:32	RSVD	R		Reserved field
31:0	DSTAT	RW	0x0	Destination status information retrieved by hardware from the address pointed to by the contents of the DSTATARx register Volatile: true

8.9.5.1.7 SSTATARx

- **Name**: Source Status Address Register for Channel x
- **Description**: After completion of each block transfer, hardware can retrieve the source status information from the user-defined address to which the contents of the SSTATARx register point. Users can select any location in system memory that would provide a 32-bit value to indicate the status of the source transfer.
- **Base Address**: 0x40013000
- **Offset**: 0x30 + x*0x58

- **Reset Value**: 0x0

Table 8-629 Source Status Address Register for Channel x

Bits	Field Name	RW	Reset	Description
63:32	RSVD	R		Reserved field
31:0	SSTATAR	RW	0x0	Pointer from where hardware can fetch the source status information which is registered in the SSTATx register and written out to the SSTATx register location of the LLI before the start of the next block.

8.9.5.1.8 DSTATARx

- **Name**: Destination Status Address Register for Channel x
- **Description**: After completion of each block transfer, hardware can retrieve the destination status information from the user-defined address to which the contents of the DSTATARx register point. Users can select any location in system memory that would provide a 32-bit value to indicate the status of the destination transfer.
- **Base Address**: 0x40013000
- **Offset**: 0x38 + x*0x58
- **Reset Value**: 0x0

Table 8-630 Destination Status Address Register for Channel x

Bits	Field Name	RW	Reset	Description
63:32	RSVD	R		Reserved field
31:0	DSTATAR	RW	0x0	Pointer from where hardware can fetch the destination status information which is registered in the DSTATx register and written out to the DSTATx register location of the LLI before the start of the next block.

8.9.5.1.9 CFGx

- **Name**: Configuration Register for Channel x
- **Description**: This register contains fields that configure the DMA transfer. The channel configuration register remains fixed for all blocks of a multi-block transfer.

Note:

You need to program this register prior to enabling the channel.

- **Base Address**: 0x40013000
- **Offset**: 0x40 + x*0x58

- **Reset Value**: 0x0000000400000e00 + (x*0x20)

Table 8-631 Configuration Register for Channel x

Bits	Field Name	RW	Reset	Description
63:47	RSVD	R		Reserved field
46:43	DEST_PER	RW	0x0	Destination hardware interface. Assigns a hardware handshaking interface to the destination of channel x if the CFG_CHx.HSG_SEL_DEST field is 0; otherwise, this field is ignored. The channel can communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. **Note**: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
42:39	SRC_PER	RW	0x0	Source hardware interface. Assigns a hardware handshaking interface to the source of channel x if the CFG_CHx.HSG_SEL_SRC field is 0; otherwise, this field is ignored. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. **Note**: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
38	SS_UPD_EN	RW	0x0	Source status update enable. Source status information is fetched only from the location pointed to by the SSTATARx register, stored in the SSTATx register and written out to the SSTATx location of the LLI, if SS_UPD_EN is high. **Value**: • 0x0 (DISABLED): Source status update is disabled. • 0x1 (ENABLED): Source status update is enabled.
37	DS_UPD_EN	RW	0x0	Destination status update enable. Destination status information is fetched only from the location pointed to by the DSTATARx register, stored in the

Bits	Field Name	RW	Reset	Description
				DSTATx register and written out to the DSTATx location of the LLI, if DS_UPD_EN is high. **Value**: • 0x0 (DISABLED): Destination status update is disabled. • 0x1 (ENABLED): Destination status update is enabled.
36:34	PROT_CTRL	RW	0x0	Protection control bits used to drive the AHB HPROT[3:1] bus. The AMBA specification recommends that the default of HPROT indicate a non-cached, non-buffered, and privileged data access. The reset value is used to indicate such an access. HPROT[0] is tied high because all transfers are data accesses, as there are no opcode fetches. There is a one-to-one mapping of these register bits to the HPROT[3:1] master interface signals. Mapping of HPROT bus is as follows: • 0x1 to HPROT[0] • CFGx.PROT_CTRL[1] to HPROT[1] • CFGx.PROT_CTRL[2] to HPROT[2] • CFGx.PROT_CTRL[3] to HPROT[3]
33	FIFO_MODE	RW	0x0	FIFO mode select. Determines how much space or data needs to be available in the FIFO before a burst transaction request is serviced. **Value**: • 0x0: Space/Data available for a single AHB transfer of the specified transfer width • 0x1: Data available is greater than or equal to half the FIFO depth for destination transfers, and space available is greater than half the FIFO depth for source transfers. The exceptions are at the end of a burst transaction request or at the end of a block transfer.
32	FLOW_CTRL_MODE	RW	0x0	Flow control mode. Determines when source transaction requests are serviced when the destination peripheral is the flow controller. **Value**:

Bits	Field Name	RW	Reset	Description
				• 0x0: Source transaction requests are serviced when they occur. Data pre-fetching is enabled. • 0x1: Source transaction requests are not serviced until a destination transaction request occurs. In this mode, the amount of data transferred from the source is limited so that it is guaranteed to be transferred to the destination prior to block termination by the destination. Data pre-fetching is disabled.
31	RELOAD_DEST	RW	0x0	Automatic destination reload. The DARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. **Value**: • 0x0: Destination reload disabled • 0x1: Destination reload enabled
30	RELOAD_SRC	RW	0x0	Automatic source reload. The SARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated. **Value**: • 0x0: Source reload disabled • 0x1: Source reload enabled
29:20	Rsvd_CFG	R	0x0	Reserved field
19	SRC_HSG_POL	RW	0x0	Source handshaking interface polarity. **Value**: • 0x0: Source handshaking interface polarity is active high. • 0x1: Source handshaking interface polarity is active low.
18	DEST_HSG_POL	RW	0x0	Destination handshaking interface polarity. **Value**: • 0x0: Destination handshaking interface polarity is active high. • 0x1: Destination handshaking interface polarity is active low.
17:12	Rsvd_CFG	R	0x0	Reserved field
11	HSG_SEL_SRC	RW	0x0	Selects software or hardware handshaking interface for source.

Bits	Field Name	RW	Reset	Description
				This register selects which of the handshaking interfaces - hardware or software - is active for source requests on this channel. If the source peripheral is memory, this bit is ignored. **Value**: • 0x0: Hardware handshaking interface. Software initiated transaction requests are ignored. • 0x1: Software handshaking interface. Hardware initiated transaction requests are ignored.
10	HSG_SEL_DEST	RW	0x0	Selects a software or hardware handshaking interface for destination. This register selects which of the handshaking interfaces - hardware or software - is active for destination requests on this channel. If the destination peripheral is memory, this bit is ignored. **Value**: • 0x0: Hardware handshaking interface. Software initiated transaction requests are ignored. • 0x1: Software handshaking interface. Hardware initiated transaction requests are ignored.
9	FIFO_EMPTY	R	0x0	Channel FIFO status. Indicates if there is data left in the channel FIFO. It can be used in conjunction with CFGx to cleanly disable a channel. **Value**: • 0x0: Channel FIFO is not empty. • 0x1: Channel FIFO is empty.
8	CH_SUSP	RW	0x0	Channel suspend. Suspends all DMA data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. It can also be used in conjunction with CFGx to cleanly disable a channel without losing any data. **Value**: • 0x0: DMA transfer from the source is not suspended. • 0x1: Suspend DMA transfer from the source.
7:5	CH_PRIOR	RW		Channel priority.

Bits	Field Name	RW	Reset	Description
				A priority of 7 is the highest priority and 0 is the lowest. This field must be programmed within the range of 0 to 3. A programmed value beyond this range will cause erroneous behavior. **Value**: • 0x0: Channel priority is 0. • 0x1: Channel priority is 1. • 0x2: Channel priority is 2. • 0x3: Channel priority is 3.
4:0	RSVD	R		Reserved field

8.9.5.1.10 SGRx

- **Name**: Source Gather Register for Channel x
- **Description**: The Source Gather Register contains two fields.
  - Source gather count field (SGRx.SGC): Specifies the number of contiguous source transfers of CTLx.SRC_TR_WIDTH between successive gather intervals. This is defined as a gather boundary.
  - Source gather interval field (SGRx.SGI): Specifies the source address increment/decrement in multiples of CTLx.SRC_TR_WIDTH on a gather boundary when gather mode is enabled for the source transfer.
- **Base Address**: 0x40013000
- **Offset**: 0x48 + x*0x58
- **Reset Value**: 0x0

Table 8-632 Source Gather Register for Channel x

Bits	Field Name	RW	Reset	Description
63:32	RSVD	R		Reserved field
31:20	SGC	RW	0x0	Source gather count.
19:0	SGI	RW	0x0	Source gather interval.

8.9.5.1.11 DSRx

- **Name**: Destination Scatter Register for Channel x
- **Description**: The Destination Status Register contains two fields.
  - Destination scatter count field (DSRx.DSC): Specifies the number of contiguous destination transfers of CTLx.DST_TR_WIDTH between successive scatter boundaries.

- Destination scatter interval field (DSRx.DSI): Specifies the destination address increment/decrement in multiples of CTLx.DST_TR_WIDTH on a scatter boundary when scatter mode is enabled for the destination transfer.

- **Base Address**: 0x40013000
- **Offset**: 0x50 + x*0x58
- **Reset Value**: 0x0

Table 8-633 Destination Scatter Register for Channel x

Bits	Field Name	RW	Reset	Description
63:32	RSVD	R		Reserved field
31:20	DSC	RW	0x0	Destination scatter count.
19:0	DSI	RW	0x0	Destination scatter interval.

8.9.5.2 Interrupt_Registers

8.9.5.2.1 INT_RSTAT_TC

- **Name**: Raw Status for Transfer Complete Interrupt
- **Description**: Interrupt events are stored in this Raw Interrupt Status Register before masking. This register has a bit allocated to each channel; for example, INT_RSTAT_TC[2] is the Channel 2 raw transfer complete interrupt. Each bit in this register is cleared by writing a 1 to the corresponding location in the INT_CLR_TC register.

Note:

Write access is available to this register or software for testing purposes only. Under normal operation, writes to this register are not recommended.

- **Base Address**: 0x40013000
- **Offset**: 0x2c0
- **Reset Value**: 0x0

Table 8-634 Raw Status for Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	Rsvd_INT_RSTAT_TC	R	0x0	Reserved field
7:0	RAW	RW	0x0	Raw status for transfer complete interrupt **Value**: • 0x0: Inactive raw interrupt status • 0x1: Active raw interrupt status

8.9.5.2.2 INT_RSTAT_BTC

- **Name**: Raw Status for Block Transfer Complete Interrupt
- **Description**: Interrupt events are stored in this Raw Interrupt Status Register before masking. This register has a bit allocated to each channel; for example, INT_RSTAT_BTC[2] is the Channel 2 raw block complete interrupt. Each bit in this register is cleared by writing a 1 to the corresponding location in the CLR_BLK register.

Note:

Write access is available to this register or software for testing purposes only. Under normal operation, writes to this register are not recommended.

- **Base Address**: 0x40013000
- **Offset**: 0x2c8
- **Reset Value**: 0x0

Table 8-635 Raw Status for Block Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_RSTAT_BTC	R	0x0	Reserved field
7:0	RAW	RW	0x0	Raw Status for Block Transfer Complete Interrupt **Value**: • 0x0: Inactive raw interrupt status • 0x1: Active raw interrupt status

8.9.5.2.3 INT_RSTAT_STC

- **Name**: Raw Status for Source Transaction Complete Interrupt
- **Description**: Interrupt events are stored in this Raw Interrupt Status Register before masking. This register has a bit allocated to each channel; for example, INT_RSTAT_STC[2] is the Channel 2 raw source transaction complete interrupt. Each bit in this register is cleared by writing a 1 to the corresponding location in the INT_CLR_STC register.

Note:

Write access is available to this register or software testing purposes only. Under normal operation, writes to this register are not recommended.

- **Base Address**: 0x40013000
- **Offset**: 0x2D0
- **Reset Value**: 0x0

Table 8-636 Raw Status for Source Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_RSTAT_STC	R	0x0	Reserved field
7:0	RAW	RW	0x0	Raw status for source transaction complete interrupt **Value**: • 0x0: Inactive raw interrupt status • 0x1: Active raw interrupt status

8.9.5.2.4 INT_RSTAT_DTC

- **Name**: Raw Status for Destination Transaction Complete Interrupt
- **Description**: Interrupt events are stored in this Raw Interrupt Status register before masking. This register has a bit allocated to each channel; for example, INT_RSTAT_DTC[2] is the Channel 2 raw destination transaction complete interrupt. Each bit in this register is cleared by writing a 1 to the corresponding location in the INT_CLR_DTC register.

Note:

Write access is available to this register or software testing purposes only. Under normal operation, writes to this register are not recommended.

- **Base Address**: 0x40013000
- **Offset**: 0x2d8
- **Reset Value**: 0x0

Table 8-637 Raw Status for Destination Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_RSTAT_DTC	R	0x0	Reserved field
7:0	RAW	RW	0x0	Raw status for destination transaction complete interrupt **Value**: • 0x0: Inactive raw interrupt status • 0x1: Active raw interrupt status

8.9.5.2.5 INT_RSTAT_ERR

- **Name**: Raw Status for Error Interrupt
- **Description**: Interrupt events are stored in this Raw Interrupt Status Register before masking. This register has a bit allocated to each channel; for example, INT_RSTAT_ERR[2] is the Channel 2 raw error interrupt. Each bit in this register is cleared by writing a 1 to the corresponding location in the INT_CLR_ERR register.

Note:

Write access is available to this register or software testing purposes only. Under normal operation, writes to this register are not recommended.

- **Base Address**: 0x40013000
- **Offset**: 0x2E0
- **Reset Value**: 0x0

Table 8-638 Raw Status for Error Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_RSTAT_ERR	R	0x0	Reserved field
7:0	RAW	RW	0x0	Raw Status for Error Interrupt **Value**: • 0x0: Inactive raw interrupt status • 0x1: Active raw interrupt status

8.9.5.2.6 INT_STAT_TC

- **Name**: Status for Transfer Complete Interrupt
- **Description**: Channel DMA transfer complete interrupt events from all channels are stored in this interrupt status register after masking. This register has a bit allocated to each channel; for example, INT_STAT_TC [2] is the Channel 2 source DMA transfer complete interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.
- **Base Address**: 0x40013000
- **Offset**: 0x2E8
- **Reset Value**: 0x0

Table 8-639 Status for Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_STAT_TC	R	0x0	Reserved field
7:0	STAT	R	0x0	Status for transfer complete interrupt **Value**: • 0x0: Inactive interrupt status • 0x1: Active interrupt status

8.9.5.2.7 INT_STAT_BTC

- **Name**: Status for Block Transfer Complete Interrupt

- **Description**: Channel block complete interrupt events from all channels are stored in this interrupt status register after masking. This register has a bit allocated to each channel; for example, INT_STAT_BTC[2] is the Channel 2 block complete interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.
- **Base Address**: 0x40013000
- **Offset**: 0x2F0
- **Reset Value**: 0x0

Table 8-640 Status for Block Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_STAT_BTC	R	0x0	Reserved field
7:0	STAT	R	0x0	Status for block transfer complete interrupt **Value**: • 0x0: Inactive interrupt status • 0x1: Active interrupt status

8.9.5.2.8 INT_STAT_STC

- **Name**: Status for Source Transaction Complete Interrupt
- **Description**: Channel source transaction complete interrupt events from all channels are stored in this interrupt status register after masking. This register has a bit allocated to each channel; for example, INT_STAT_STC[2] is the Channel 2 source transaction complete interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.
- **Base Address**: 0x40013000
- **Offset**: 0x2F8
- **Reset Value**: 0x0

Table 8-641 Status for Source Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_STAT_STC	R	0x0	Reserved field
7:0	STAT	R	0x0	Status for source transaction complete interrupt **Value**: • 0x0: Inactive interrupt status • 0x1: Active interrupt status

8.9.5.2.9 INT_STAT_DTC

- **Name**: Status for Destination Transaction Complete Interrupt

- **Description**: Channel destination transaction complete interrupt events from all channels are stored in this interrupt status register after masking. This register has a bit allocated to each channel; for example, INT_STAT_DTC[2] is the Channel 2 status destination transaction complete interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.
- **Base Address**: 0x40013000
- **Offset**: 0x300
- **Reset Value**: 0x0

Table 8-642 Status for Destination Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_STAT_DTC	R	0x0	Reserved field
7:0	STAT	R	0x0	Status for destination transaction complete interrupt **Value**: • 0x0: Inactive interrupt status • 0x1: Active interrupt status

8.9.5.2.10 INT_STAT_ERR

- **Name**: Status for Error Interrupt
- **Description**: Channel error interrupt events from all channels are stored in this interrupt status register after masking. This register has a bit allocated to each channel; for example, INT_STAT_ERR[2] is the Channel 2 status error interrupt. The contents of this register are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.
- **Base Address**: 0x40013000
- **Offset**: 0x308
- **Reset Value**: 0x0

Table 8-643 Status for Error Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_STAT_ERR	R	0x0	Reserved field
7:0	STAT	R	0x0	Status for error interrupt **Value**: • 0x0: Inactive interrupt status • 0x1: Active interrupt status

8.9.5.2.11 INT_MASK_TC

- **Name**: MASK for Transfer Complete Interrupt

- **Description**: The contents of the raw status register INT_RSTAT_TC are masked with the contents of the mask register INT_MASK_TC. This register has a bit allocated to each channel; for example, INT_MASK_TC[2] is the mask bit for the Channel 2 transfer complete interrupt. A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same AHB write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the INT_MASK_TC register writes a 1 into INT_MASK_TC[0], while INT_MASK_TC[7:1] remains unchanged. Writing hex 00xx leaves INT_MASK_TC[7:0] unchanged. Writing a 1 to any bit in this register unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the status registers and int_port signals.
- **Base Address**: 0x40013000
- **Offset**: 0x310
- **Reset Value**: 0x0

Table 8-644 MASK for Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:16	RSVD_INT_MASK_TC	R	0x0	Reserved field
15:8	INT_MASK_WE	W	0x0	Interrupt Mask Write Enable **Value**: • 0x0: Interrupt mask write disabled • 0x1: Interrupt mask write enabled
7:0	INT_MASK	RW	0x0	Mask for Transfer Complete Interrupt **Value**: • 0x0: Mask the interrupts. • 0x1: Unmask the interrupts.

8.9.5.2.12 INT_MASK_BTC

- **Name**: Mask for Block Transfer Complete Interrupt
- **Description**: The contents of the raw status register INT_RSTAT_BTC are masked with the contents of the mask register INT_MASK_BTC. This register has a bit allocated to each channel; for example, INT_MASK_BTC[2] is the mask bit for the Channel 2 block complete interrupt. A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same AHB write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the INT_MASK_BTC register writes a 1 into INT_MASK_BTC[0], while INT_MASK_BTC[7:1] remains unchanged. Writing hex 00xx leaves INT_MASK_BTC[7:0] unchanged. Writing a 1 to any bit in this register unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the status registers and int_port signals.
- **Base Address**: 0x40013000

- **Offset**: 0x318
- **Reset Value**: 0x0

Table 8-645 Mask for Block Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:16	RSVD_INT_MASK_BTC	R	0x0	Reserved field
15:8	INT_MASK_WE	W	0x0	Interrupt Mask Write Enable **Value**: • 0x0: Interrupt mask write disabled • 0x1: Interrupt mask write enabled
7:0	INT_MASK	RW	0x0	Mask for Block Transfer Complete Interrupt **Value**: • 0x0: Mask the interrupts. • 0x1: Unmask the interrupts.

8.9.5.2.13 INT_MASK_STC

- **Name**: MASK for Source Transaction Complete Interrupt
- **Description**: The contents of the raw status register INT_RSTAT_STC are masked with the contents of the mask register INT_MASK_STC. This register has a bit allocated to each channel; for example, INT_MASK_STC[2] is the mask bit for the Channel 2 source transaction complete interrupt. When the source peripheral of DMA channel i is memory, then the source transaction complete interrupt, INT_MASK_STC[i], must be masked to prevent an erroneous triggering of an interrupt on the int_combined signal. A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same AHB write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the INT_MASK_STC register writes a 1 into INT_MASK_STC[0], while INT_MASK_STC[7:1] remains unchanged. Writing hex 00xx leaves INT_MASK_STC[7:0] unchanged. Writing a 1 to any bit in this register unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the status registers and int_port signals.
- **Base Address**: 0x40013000
- **Offset**: 0x320
- **Reset Value**: 0x0

Table 8-646 MASK for Source Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:16	RSVD_INT_MASK_STC	R	0x0	Reserved field
15:8	INT_MASK_WE	W	0x0	Interrupt Mask Write Enable **Value**:

Bits	Field Name	RW	Reset	Description
				• 0x0: Interrupt mask write disabled • 0x1: Interrupt mask write enabled
7:0	INT_MASK	RW	0x0	Mask for Source Transaction Complete Interrupt **Value**: • 0x0: Mask the interrupts. • 0x1: Unmask the interrupts.

8.9.5.2.14 INT_MASK_DTC

- **Name**: Mask for Destination Transaction Complete Interrupt
- **Description**: The contents of the raw status register INT_RSTAT_DTC are masked with the contents of the mask register INT_MASK_DTC. This register has a bit allocated to each channel; for example, INT_MASK_DTC[2] is the mask bit for the Channel 2 destination transaction complete interrupt. When the destination peripheral of DMA channel i is memory, then the destination transaction complete interrupt, INT_MASK_DTC[i], must be masked to prevent an erroneous triggering of an interrupt on the int_combined(_n) signal. A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same AHB write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the INT_MASK_DTC register writes a 1 into INT_MASK_DTC[0], while INT_MASK_DTC[7:1] remains unchanged. Writing hex 00xx leaves INT_MASK_DTC[7:0] unchanged. Writing a 1 to any bit in this register unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the status registers and int_port signals.
- **Base Address**: 0x40013000
- **Offset**: 0x328
- **Reset Value**: 0x0

Table 8-647 Mask for Destination Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:16	RSVD_INT_MASK_DTC	R	0x0	Reserved field
15:8	INT_MASK_WE	W	0x0	Interrupt Mask Write Enable **Value**: • 0x0: Interrupt mask write disabled • 0x1: Interrupt mask write enabled
7:0	INT_MASK	RW	0x0	Mask for Destination Transaction Complete Interrupt **Value**: • 0x0: Mask the interrupts. • 0x1: Unmask the interrupts.

8.9.5.2.15 INT_MASK_ERR

- **Name**: Mask for Error Interrupt
- **Description**: The contents of the raw status register INT_RSTAT_ERR are masked with the contents of the mask register INT_MASK_ERR. This register has a bit allocated to each channel; for example, INT_MASK_ERR[2] is the mask bit for the Channel 2 error interrupt. A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same AHB write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the INT_MASK_ERR register writes a 1 into INT_MASK_ERR[0], while INT_MASK_ERR[7:1] remains unchanged. Writing hex 00xx leaves INT_MASK_ERR[7:0] unchanged. Writing a 1 to any bit in this register unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the Status registers and int_port signals.
- **Base Address**: 0x40013000
- **Offset**: 0x330
- **Reset Value**: 0x0

Table 8-648 Mask for Error Interrupt

Bits	Field Name	RW	Reset	Description
63:16	RSVD_INT_MASK_ERR	R	0x0	Reserved field
15:8	INT_MASK_WE	W	0x0	Interrupt Mask Write Enable **Value**: • 0x0: Interrupt mask write disabled • 0x1: Interrupt mask write enabled
7:0	INT_MASK	RW	0x0	Mask for Error Interrupt **Value**: • 0x0: Mask the interrupts. • 0x1: Unmask the interrupts.

8.9.5.2.16 INT_CLR_TC

- **Name**: Clear for Transfer Complete Interrupt
- **Description**: Each bit in the INT_RSTAT_TC and INT_STAT_TC is cleared on the same cycle by writing a 1 to the corresponding location in the registers. This register has a bit allocated to each channel; for example, INT_CLR_TC[2] is the clear bit for the Channel 2 transfer done interrupt. Writing a 0 has no effect. This register is not readable.
- **Base Address**: 0x40013000
- **Offset**: 0x338
- **Reset Value**: 0x0

Table 8-649 Clear for Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_CLR_TC	W	0x0	Reserved field
7:0	CLR	W	0x0	Clear for Transfer Complete Interrupt **Value**: • 0x0: No effect • 0x1: Clears interrupts.

8.9.5.2.17 INT_CLR_BTC

- **Name**: Clear for Block Transfer Complete Interrupt
- **Description**: Each bit in the INT_RSTAT_BTC and INT_STAT_BTC is cleared on the same cycle by writing a 1 to the corresponding location in the registers. This register has a bit allocated to each channel; for example, INT_CLR_BTC[2] is the clear bit for the Channel 2 block done interrupt. Writing a 0 has no effect. This register is not readable.
- **Base Address**: 0x40013000
- **Offset**: 0x340
- **Reset Value**: 0x0

Table 8-650 Clear for Block Transfer Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_CLR_BTC	W	0x0	Reserved field
7:0	CLR	W	0x0	Clear for Block Transfer Complete Interrupt

8.9.5.2.18 INT_CLR_STC

- **Name**: Clear for Source Transaction Complete Interrupt
- **Description**: Each bit in the INT_RSTAT_STC and INT_STAT_STC is cleared on the same cycle by writing a 1 to the corresponding location in the registers. This register has a bit allocated to each channel; for example, INT_CLR_STC[2] is the clear bit for the Channel 2 source transaction done interrupt. Writing a 0 has no effect. This register is not readable.
- **Base Address**: 0x40013000
- **Offset**: 0x348
- **Reset Value**: 0x0

Table 8-651 Clear for Source Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_CLR_STC	W	0x0	Reserved field
7:0	CLR	W	0x0	Clear for Source Transaction Complete Interrupt **Value**: • 0x0: No effect • 0x1: Clears interrupts.

8.9.5.2.19 INT_CLR_DTC

- **Name**: Clear for Destination Transaction Complete Interrupt
- **Description**: Each bit in the INT_RSTAT_DTC and INT_STAT_DTC is cleared on the same cycle by writing a 1 to the corresponding location in the registers. This register has a bit allocated to each channel; for example, INT_CLR_DTC[2] is the clear bit for the Channel 2 destination transaction done interrupt. Writing a 0 has no effect. This register is not readable.
- **Base Address**: 0x40013000
- **Offset**: 0x350
- **Reset Value**: 0x0

Table 8-652 Clear for Destination Transaction Complete Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_CLR_DTC	W	0x0	Reserved field
7:0	CLR	W	0x0	Clear for Destination Transaction Complete Interrupt **Value**: • 0x0: No effect • 0x1: Clears interrupts.

8.9.5.2.20 INT_CLR_ERR

- **Name**: Clear for Error Interrupt
- **Description**: Each bit in the INT_RSTAT_ERR and INT_STAT_ERR is cleared on the same cycle by writing a 1 to the corresponding location in the registers. This register has a bit allocated to each channel; for example, INT_CLR_ERR[2] is the clear bit for the Channel 2 error interrupt. Writing a 0 has no effect. This register is not readable.
- **Base Address**: 0x40013000
- **Offset**: 0x358
- **Reset Value**: 0x0

Table 8-653 Clear for Error Interrupt

Bits	Field Name	RW	Reset	Description
63:8	RSVD_INT_CLR_ERR	W	0x0	Reserved field
7:0	CLR	W	0x0	Clear for Error Interrupt **Value**: • 0x0: No effect • 0x1: Clears interrupts.

8.9.5.2.21 INT_STAT_ET

- **Name**: Status for Each Interrupt Type
- **Description**: The contents of each of the five status registers INT_STAT_TC, INT_STAT_BTC, INT_STAT_STC, INT_STAT_DTC, and INT_STAT_ERR are ORed to produce a single bit for each interrupt type in the combined status register (INT_STAT_ET). This register is read-only.
- **Base Address**: 0x40013000
- **Offset**: 0x360
- **Reset Value**: 0x0

Table 8-654 Status for Each Interrupt Type

Bits	Field Name	RW	Reset	Description
63:5	RSVD_INT_STAT_ET	R	0x0	Reserved field
4	ERR	R	0x0	OR of the contents of INT_STAT_ERR **Value**: • 0x0: OR of the contents of INT_STAT_ERR register is 0. • 0x1: OR of the contents of INT_STAT_ERR register is 1.
3	DEST_XFE_CPLT	R	0x0	OR of the contents of INT_STAT_DTC **Value**: • 0x0: OR of the contents of INT_STAT_DTC register is 0. • 0x1: OR of the contents of INT_STAT_DTC register is 1.
2	SRC_XFE_CPLT	R	0x0	OR of the contents of INT_STAT_STC **Value**: • 0x0: OR of the contents of INT_STAT_STC register is 0. • 0x1: OR of the contents of INT_STAT_STC register is 1.
1	BLK_XFE_CPLT	R	0x0	OR of the contents of INT_STAT_BTC register **Value**: • 0x0: OR of the contents of INT_STAT_BTC register is 0.

Bits	Field Name	RW	Reset	Description
				• 0x1: OR of the contents of INT_STAT_BTC register is 1.
0	XFE_CPLT	R	0x0	OR of the contents of INT_STAT_TC register **Value**: • 0x0: OR of the contents of INT_STAT_TC register is 0. • 0x1: OR of the contents of INT_STAT_TC register is 1.

8.9.5.3 Software_Handshake_Registers

8.9.5.3.1 REQ_SST

- **Name**: Source Software Transaction Request Register
- **Description**: A bit is assigned for each channel in this register. REQ_SST[n] is ignored when software handshaking is not enabled for the source of channel n. A channel SRC_REQ bit is written only if the corresponding channel write enable bit in the SRC_REQ_WE field is asserted on the same AHB write transfer, and if the channel is enabled in the CH_EN register. For example, writing hex 0101 writes a 1 into REQ_SST[0], while REQ_SST[7:1] remains unchanged. Writing hex 00xx leaves REQ_SST[7:0] unchanged. This allows software to set a bit in the REQ_SST register without performing a read-modified write operation.
- **Base Address**: 0x40013000
- **Offset**: 0x368
- **Reset Value**: 0x0

Table 8-655 Source Software Transaction Request Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_REQ_SST	R	0x0	Reserved field
15:8	SRC_REQ_WE	RW	0x0	Source Software Transaction Request write enable **Value**: • 0x0: Source request write disabled • 0x1: Source request write enabled
7:0	SRC_REQ	RW	0x0	Source Software Transaction Request **Value**: • 0x0: Source request is not active. • 0x1: Source request is active.

8.9.5.3.2 REQ_DST

- **Name**: Destination Software Transaction Request Register

- **Description**: A bit is assigned for each channel in this register. REQ_DST[n] is ignored when software handshaking is not enabled for the source of channel n. A channel DEST bit is written only if the corresponding channel write enable bit in the DEST_REQ_WE field is asserted on the same AHB write transfer, and if the channel is enabled in the CH_EN register.
- **Base Address**: 0x40013000
- **Offset**: 0x370
- **Reset Value**: 0x0

Table 8-656 Destination Software Transaction Request Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_REQ_DST	R	0x0	Reserved field
15:8	DEST_REQ_WE	RW	0x0	Destination Software Transaction Request write enable **Value**: • 0x0: Destination request write disabled • 0x1: Destination request write enabled
7:0	DEST_REQ	RW	0x0	Destination Software Transaction Request **Value**: • 0x0: Destination request is not active. • 0x1: Destination request is active.

8.9.5.3.3 REQ_SGL_ST

- **Name**: Source Single Transaction Request Register
- **Description**: A bit is assigned for each channel in this register. REQ_SGL_ST is ignored when software handshaking is not enabled for the source of channel n. A channel SRC_SGL_REQ bit is written only if the corresponding channel write enable bit in the SRC_SGL_REQ_WE field is asserted on the same AHB write transfer, and if the channel is enabled in the CH_EN register.
- **Base Address**: 0x40013000
- **Offset**: 0x378
- **Reset Value**: 0x0

Table 8-657 Source Single Transaction Request Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_REQ_SGL_ST	R	0x0	Reserved field
15:8	SRC_SGL_REQ_WE	RW	0x0	Source Single Transaction Request write enable **Value**: • 0x0: Single write disabled

Bits	Field Name	RW	Reset	Description
				• 0x1: Single write enabled
7:0	SRC_SGL_REQ	RW	0x0	Source Single Transaction Request **Value**: • 0x0: Source request is not active. • 0x1: Source request is active.

8.9.5.3.4 REQ_SGL_DT

- **Name**: Destination Single Transaction Request Register
- **Description**: A bit is assigned for each channel in this register. REQ_SGL_DT is ignored when software handshaking is not enabled for the destination of channel n. A channel DEST_SGL_REQ bit is written only if the corresponding channel write enable bit in the DST_SGL_REQ_WE field is asserted on the same AHB write transfer, and if the channel is enabled in the CH_EN register.
- **Base Address**: 0x4A0013000
- **Offset**: 0x380
- **Reset Value**: 0x0

Table 8-658 Destination Single Transaction Request Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_REQ_SGL_DT	R	0x0	Reserved field
11:8	DST_SGL_REQ_WE	RW	0x0	Destination Single Transaction Request write enable **Value**: • 0x0: Destination write disabled • 0x1: Destination write enabled
7:0	DEST_SGL_REQ	RW	0x0	Destination Single Transaction Request **Value**: • 0x0: Destination single or burst request is not active. • 0x1: Destination single or burst request is active.

8.9.5.3.5 REQ_LST_ST

- **Name**: Source Last Transaction Request Register
- **Description**: A bit is assigned for each channel in this register. REQ_LST_ST is ignored when software handshaking is not enabled for the source of channel n, or when the source of channel n is not a flow controller. A channel LST_SRC bit is written only if the corresponding channel write enable bit in the LST_SRC_WE field is asserted on the same AHB write transfer, and if the channel is enabled in the CH_EN register.
- **Base Address**: 0x40013000

- **Offset**: 0x388
- **Reset Value**: 0x0

Table 8-659 Source Last Transaction Request Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_REQ_LST_ST	R	0x0	Reserved field
15:8	LST_SRC_WE	RW	0x0	Source Last Transaction Request write enable **Value**: • 0x0: Source last transaction request write disabled • 0x1: Source last transaction request write enabled
7:0	LST_SRC	RW	0x0	Source Last Transaction Request register **Value**: • 0x0: Not last transaction in current block • 0x1: Last transaction in current block

8.9.5.3.6 REQ_LST_DT

- **Name**: Destination Last Transaction Request Register
- **Description**: A bit is assigned for each channel in this register. REQ_LST_DT is ignored when software handshaking is not enabled for the destination of channel n or when the destination of channel n is not a flow controller. A channel LST_DEST bit is written only if the corresponding channel write enable bit in the LST_DEST_WE field is asserted on the same AHB write transfer, and if the channel is enabled in the CH_EN register.
- **Base Address**: 0x40013000
- **Offset**: 0x390
- **Reset Value**: 0x0

Table 8-660 Destination Last Transaction Request Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_REQ_LST_DT	R	0x0	Reserved field
15:8	LST_DEST_WE	RW	0x0	Destination Last Transaction Request write enable **Value**: • 0x0: Destination last transaction request write disabled • 0x1: Destination last transaction request write enabled
7:0	LST_DEST	RW	0x0	Destination Last Transaction Request **Value**: • 0x0: Not last transaction in current block

Bits	Field Name	RW	Reset	Description
				• 0x1: Last transaction in current block

8.9.5.4 Miscellaneous_Registers

8.9.5.4.1 CFG

- **Name**: DMA Configuration Register
- **Description**: This register is used to enable the DMA, which must be done before any channel activity. If the global channel enable bit is cleared while any channel is still active, CFG.DMA_EN still returns 1 to indicate that there are channels still active until hardware has terminated all activities on all channels, at which point the CFG.DMA_EN bit returns 0.
- **Base Address**: 0x40013000
- **Offset**: 0x398
- **Reset Value**: 0x0

Table 8-661 DMA Configuration Register

Bits	Field Name	RW	Reset	Description
63:1	RSVD_CFG	R	0x0	Reserved field
0	DMA_EN	RW	0x0	DMA Enable bit. **Value**: • 0x0: DMA disabled • 0x1: DMA enabled

8.9.5.4.2 CH_EN

- **Name**: DMA Channel Enable Register
- **Description**: This is the DMA Channel Enable Register. If software needs to set up a new channel, it can read this register in order to find out which channels are currently inactive; it can then enable an inactive channel with the required priority. All bits of this register are cleared to 0 when the global DMA channel enable bit, CFG[0], is 0. When the global channel enable bit is 0, a write to the CH_EN register is ignored and a read will always read back 0. The channel enable bit, CH_EN.CH_EN, is written only if the corresponding channel write enable bit, CH_EN.CH_EN_WE, is asserted on the same AHB write transfer. For example, writing hex 01x1 writes a 1 into CH_EN[0], while CH_EN[7:1] remains unchanged. Writing hex 00xx leaves CH_EN[7:0] unchanged. Note that a read-modified write is not required.
- **Base Address**: 0x40013000
- **Offset**: 0x3a0
- **Reset Value**: 0x0

Table 8-662 DMA Channel Enable Register

Bits	Field Name	RW	Reset	Description
63:16	Rsvd_1_CH_EN	R	0x0	Reserved field
15:8	CH_EN_WE	W	0x0	Channel enable register
7:0	CH_EN	RW	0x0	Channel Enable. The CH_EN.CH_EN bit is automatically cleared by hardware to disable the channel after the last AMBA transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer. **Value**: • 0x0: Disable the channel. • 0x1: Enable the channel.

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

9 Communication Subsystem

9.1 Supported Features

GR5526 has a communication subsystem that supports Bluetooth Low Energy 5.3.

9.2 Transceiver

The transceiver consists of a receiver path with Low Noise Amplifier (LNA), Mixer, Baseband (BB) amplifier, and an Analog-to-Digital Converter (ADC).

The digitized signals are sent to the digital frontend (DFE) for demodulation. The DFE provides AGC feedback signals to adjust the gain of the LNA and BB amplifier to maximize the SNR at the demodulation.

On the transmit side, the digital signal from the DFE is used to modulate the CPLL_192M and delivers the modulated carrier to the PA which is configurable by the digital gain settings to deliver the required output power.

RF frequency and digital clocks are generated from the HFXO_32M.

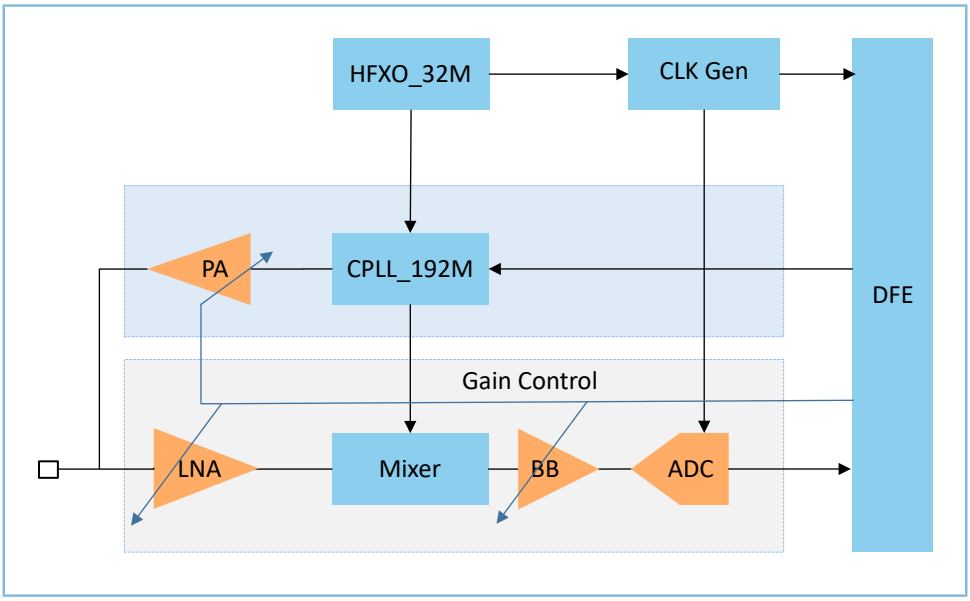

Figure 9-1 Transceiver architecture

9.3 Digital Bluetooth LE Subsystem

The digital side of Bluetooth LE subsystem is shown in Figure 9-2.

On the receive side, the digitized samples from the transceiver are filtered by a digital filter to remove the unwanted blockers, and the RSSI information is extracted. Based on the signal strength, a feedback signal is sent back to the transceiver to adjust the gain of the LNA and BB accordingly. The modem detects the frame start using the preamble, and the frequency demodulator extracts the bits inside the Bluetooth LE packet. Bit stream processing block checks the CRC of the packet content and writes it to the packet memory along with any control or status information (RSSI and RF Channel). The crypto engine decrypts the packet if it is encrypted using AES-128. The MAC firmware running on the SoC's MCU gets the packet information and processes it accordingly.

On the transmit side, the MAC firmware prepares the data to be transmitted and maps it to an event with a specific time stamp according to its connection. The Event Scheduler sets the event time and configures the packet controller to fill the data into packets and writes it to the packet memory along with its control information. The memory controller sends the packet from the memory to the modem and configures the RF channel frequency for the data packets using the packet control parameters. The packet is encrypted; CRC is added and modulated, and then sent to the frontend that passes it through a Gaussian filter and hands it to the RF transceiver.

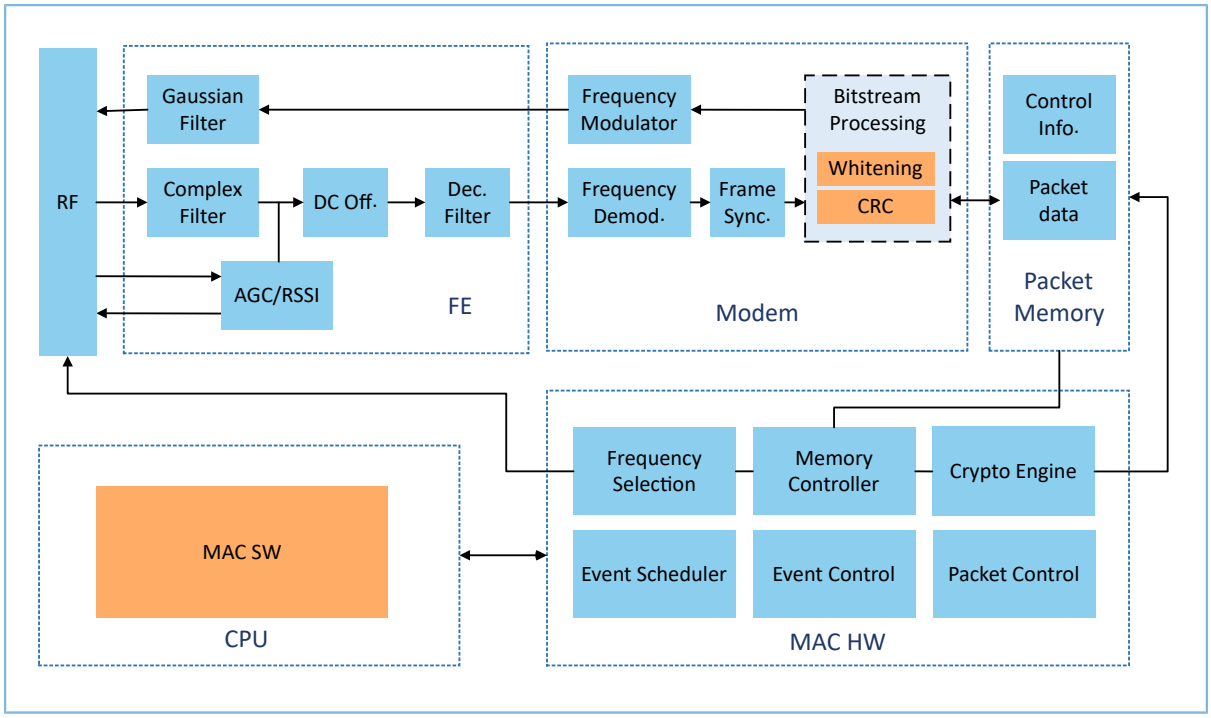

Figure 9-2 Digital Bluetooth LE subsystem architecture

9.4 Performance

9.4.1 General Performance

Table 9-1 General performance

Parameter	BLE Spec.	Min.	Typ.	Max.	Unit
Frequency	N/A	2402	-	2480	MHz
RSSI resolution	N/A	-	1	-	dBm

9.4.2 Receiver Performance

Table 9-2 Receiver performance

Parameter			BLE Spec.	Min.	Typ.	Max.	Unit
Sensitivity	1 Mbps Bluetooth LE mode	Packet length 37 bytes, dirty off	≤ -70	-99	-98	-97	dBm
		Packet length 255 bytes, dirty on	≤ -70	-96.5	-95	-93	dBm
	2 Mbps Bluetooth LE mode	Packet length 37 bytes, dirty off	≤ -70	-95.5	-94	-93	dBm
	LR S2 (125 kbps) Bluetooth LE mode	Packet length 37 bytes, dirty off	≤ -70	-101.5	-101	-100	dBm
	LR S8 (125 kbps) Bluetooth LE mode	Packet length 37 bytes, dirty off	≤ -70	-105	-104	-101	dBm
Maximum receive signal level			≥ -10	-	-2	-	dBm
Intermodulation (IMD)	1 Mbps Bluetooth LE mode	IMD performance, n = 5, packet length 37 bytes	≥ -50	-35	-34	-32	dBm
	2 Mbps Bluetooth LE mode	IMD performance, n = 5, packet length 37 bytes	≥ -50	-37	-35	-34	dBm
Selectivity	CCI/ACI @ 1 Mbps Bluetooth LE mode	Co-channel interference	≤ 21	8	8	8	dB
		Adjacent channel (+1 MHz) interference	≤ 15	-12	-11	-11	dB
		Adjacent channel (-1 MHz, image frequency + 1 MHZ) interference	≤ -15	-4	-4	-4	dB
		Adjacent channel (+2 MHz, image frequency) interference	≤ -17	-39	-38	-38	dB
		Adjacent channel (-2 MHz, image frequency) interference	≤ -9	-24	-23	-23	dB
		Adjacent channel (≥ +3 MHz) interference	≤ -27	-50	-47	-40	dB
		Adjacent channel (≤-3 MHz) interference	≤ -27	-44	-40	-36	dB
	CCI/ACI @2 Mbps Bluetooth LE mode	Co-channel interference	≤ 21	8	8	8	dB
		Adjacent channel (+2 MHz) interference	≤ 15	-13	-12	-12	dB
		Adjacent channel (-2 MHz, image frequency +2 MHz) interference	≤ 15	-5	-5	-5	dB

[8] *The typical sensitivity is the average value across all channels with RF PRIOR_RF_RX_PERFORMANCE settings. If with RF PRIOR_RF_RX_POWER (default) settings, the sensitivity would degrade 0.6 dBm. Refer to the API ble_rf_rx_performance_selection for this settings.*

Parameter			BLE Spec.	Min.	Typ.	Max.	Unit
		Adjacent channel (+4 MHz) interference	≤ -17	-42	-42	-42	dB
		Adjacent channel (-4 MHz, image frequency) interference	≤ -9	-24	-24	-23	dB
		Adjacent channel (≥ +6 MHz) interference	≤ -27	-48	-41	-34	dB
		Adjacent channel ((≤ -6 MHz) interference	≤ -27	-44	-40	-36	dB
	CCI/ACI @LE coded (S=2)	Co-channel interference	≤ 17	5	5	5	dB
		Adjacent channel (+1 MHz) interference	≤ 11	-13	-13	-13	dB
		Adjacent channel (-1MHz, image frequency + 1 MHZ) interference	≤ 11	-8	-8	-8	dB
		Adjacent channel (+2 MHz) interference	≤ -21	-42	-41	-41	dB
		Adjacent channel (-2 MHz, image frequency) interference	≤ -13	-28	-28	-27	dB
		Adjacent channel (≥ +3 MHz) interference	≤ -31	-53	-51	-47	dB
		Adjacent channel (≤ -3 MHz) interference	≤ -31	-48	-42	-38	dB
	CCI/ACI @LE coded (S=8)	Co-channel interference	≤ 12	4	4	4	dB
		Adjacent channel (+1 MHz) interference	≤ 6	-19	-19	-19	dB
		Adjacent channel (-1 MHz, image frequency + 1MHZ) interference	≤ 6	-9	-9	-9	dB
		Adjacent channel (+2 MHz) interference	-26	-40	-39	-39	dB
		Adjacent channel (-2 MHz, image frequency) interference	≤ -18	-29	-28	-28	dB
		Adjacent channel (≥ +3 MHz) interference	≤ -36	-51	-47	-43	dB
		Adjacent channel (≤-3 MHz) interference	≤ -36	-45	-41	-38	dB

Parameter			BLE Spec.	Min.	Typ.	Max.	Unit
Blocking	Interference signal frequency @ 1 Mbps Bluetooth LE mode	30 MHz–2000 MHz	≥ -30	-	-25	-	dBm
		2003 MHz–2399 MHz	≥ -35	-	-25	-	dBm
		2484 MHz–2997 MHz	≥ -35	-	-25	-	dBm
		3000 MHz–12750 MHz	≥ -30	-	-25	-	dBm
	Interference signal frequency @ 2 Mbps Bluetooth LE mode	30 MHz–2000 MHz	≥ -30	-	-25	-	dBm
		2003 MHz–2399 MHz	≥ -35	-	-25	-	dBm
		2484 MHz–2997 MHz	≥ -35	-	-25	-	dBm
		3000 MHz–12750 MHz	≥ -30	-	-25	-	dBm

[8] *The typical sensitivity is the average value across all channels with RF PRIOR_RF_RX_PERFORMANCE settings. If with RF PRIOR_RF_RX_POWER (default) settings, the sensitivity would degrade 0.6 dBm. Refer to the API ble_rf_rx_performance_selection for this settings.*

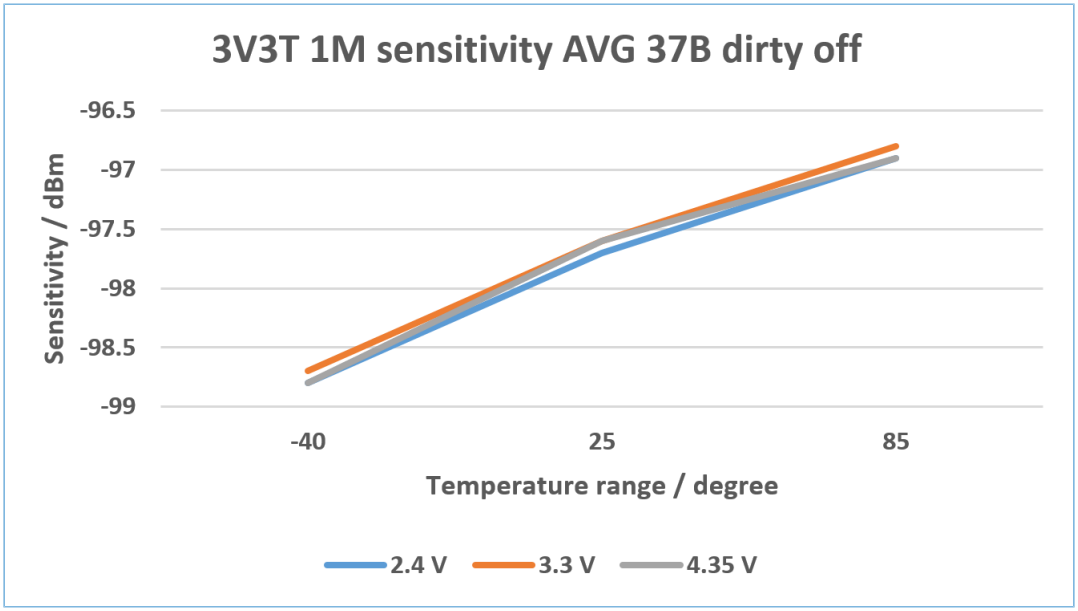

Figure 9-3 3V3T 1M sensitivity AVG 37B dirty off

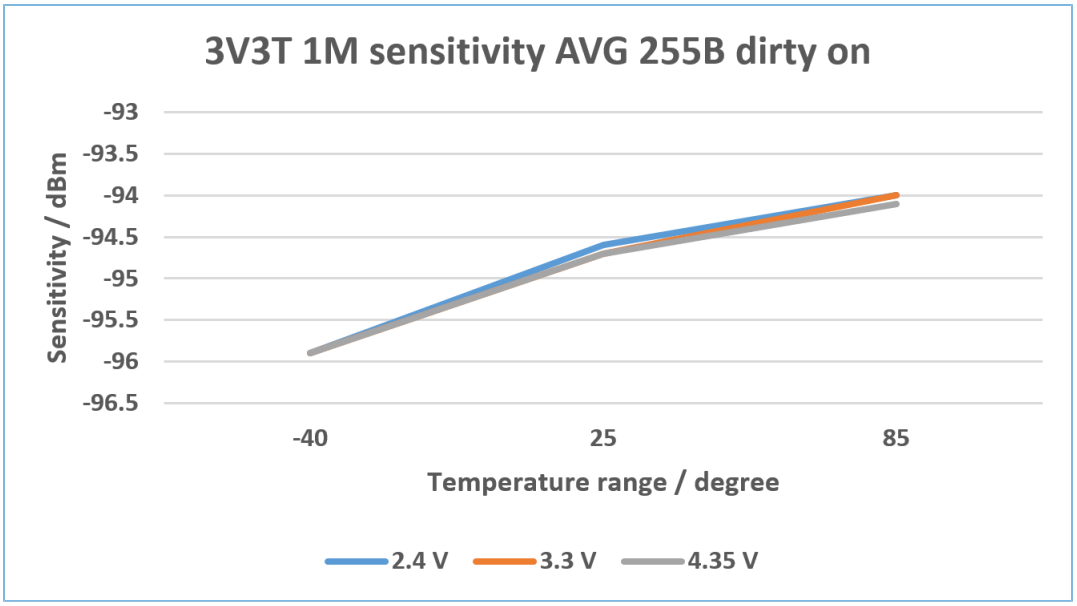

Figure 9-4 3V3T 1M sensitivity AVG 255B dirty on

9.4.3 Transmitter Performance

Table 9-3 Transmitter performance

Parameter			BLE Spec.	Min.	Typ.	Max.	Unit
Power		Output power range	≤ 20	-20	-	7	dBm
		P_{pk}-P_{avg}	≤ 3	0.44	0.46	0.49	dB
		Output power accuracy	N/A	-1.5	1.5	1.5	dBm
Modulation	1 Mbps Bluetooth LE mode	$\Delta f1_{avg}$ modulation	225–275	246	248	249	kHz
		99% df2 max	≥ 185	205	215	220	kHz
		$\Delta f2_{avg}/\Delta f1_{avg}$	≥ 0.8	0.94	0.95	0.96	
	2 Mbps Bluetooth LE mode	$\Delta f1_{avg}$ modulation	450–550	498	501	505	kHz
		99% df2 max	≥ 370	417	419	421	kHz
		$\Delta f2_{avg}/\Delta f1_{avg}$	≥ 0.8	0.93	0.94	0.95	
In-band spurious emission	1 Mbps Bluetooth LE mode	In-band spurious emission (±2 MHz offset)	≤ -20	-40	-33	-31	dBm
		In-band spurious emission (≥ ±3 MHz offset)	≤ -30	-43	-40	-39	dBm
	2 Mbps Bluetooth LE mode	In-band spurious emission (±4 MHz offset)	≤ -20	-45	-44	-43	dBm
		In-band spurious emission (±5 MHz offset)	≤ -20	-49	-47	-45	dBm
		In-band spurious emission (≥ ±6 MHz offset)	≤ -30	-51	-48	-46	dBm
Harmonic		2nd harmonic P_{out} @ 7dBm TX power	N/A	-	-35	-	dBm

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

Parameter			BLE Spec.	Min.	Typ.	Max.	Unit
		3rd harmonic P_{out} @ 7 dBm TX power	N/A	-	-53	-	dBm
		4th harmonic P_{out} @ 7 dBm TX power	N/A	-	-67	-	dBm
		5th harmonic P_{out} @ 7 dBm TX power	N/A	-	-72	-	dBm
Carrier frequency offset and drift		Carrier frequency offset ($f_n - f_{tx}$)	±150	±5	±20	±40	kHz
		Frequency drift ($f_0 - f_n$)	±50	±5	±15	±25	kHz
		Initial frequency drift ($f_1 - f_0$)	±23	±0.5	±5	±12	kHz
		Maximum drift rate ($f_n - f_{n-5}$)	±20	±0.5	±5	±10	kHz/50 µs

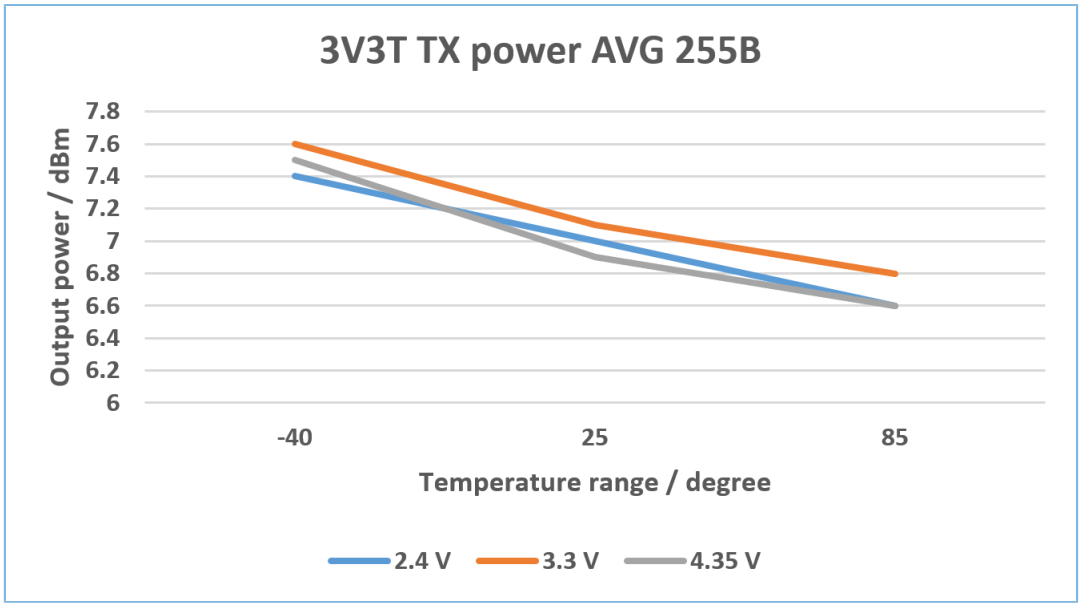

Figure 9-5 3V3T TX power AVG 255B

10 Package Information

GR5526 offers BGA83 and QFN68 packages to support different environmental requirements.

10.1 BGA83

GR5526 BGA83, including GR5526VGBIP BGA83 and GR5526VGBI BGA83, is an 83-pin and 4.3 x 4.3 x 0.96 (mm) package. It is qualified to MSL 3.

Table 10-1 BGA83 package information

Parameter	Value	Unit	Tolerance
Package Size	4.3 x 4.3	mm	±0.1 mm
BGA Ball Count	83		
Total Thickness	0.96	mm	±0.1 mm
BGA Ball Pitch	0.40		
Ball Diameter	0.20		±0.05 mm
Ball Height	0.14		±0.05 mm

The figure below shows the BGA83 package outlines.

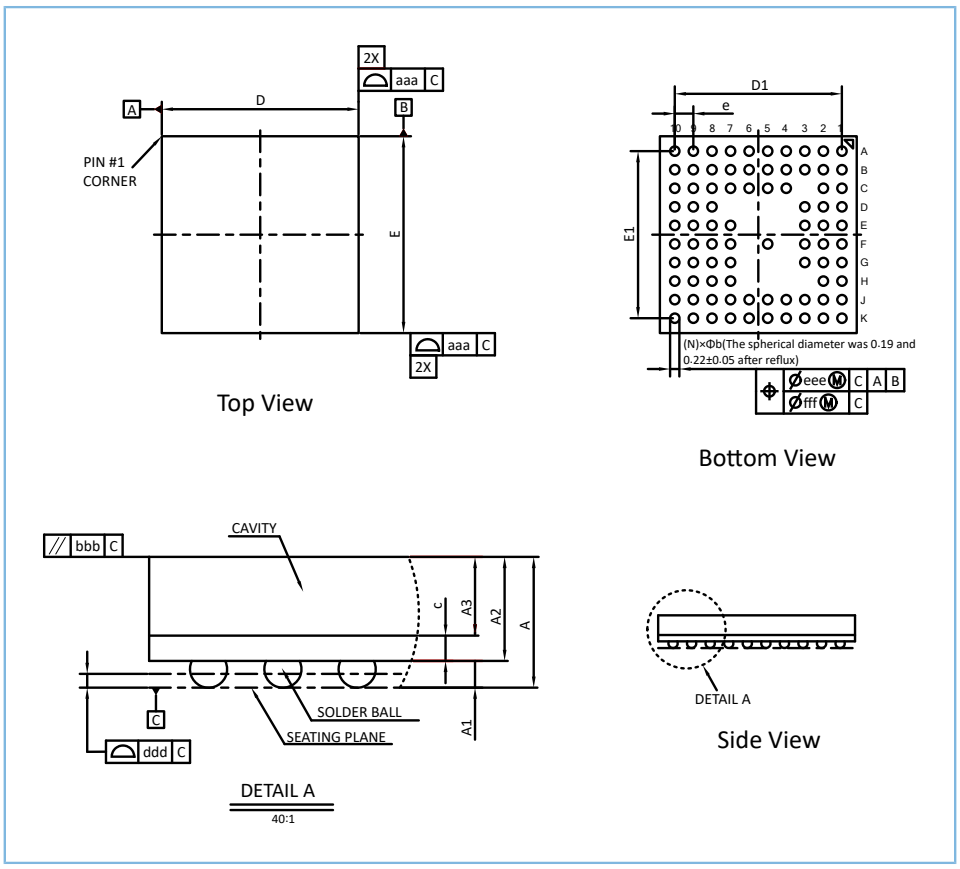

Figure 10-1 BGA83 package outlines

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd.

Note:

Drawing is not to scale.

Table 10-2 BGA83 package dimensions

Symbol	Dimensions in mm			Dimensions in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.860	0.960	1.060	0.034	0.038	0.042
A1	0.090	0.140	0.190	0.004	0.006	0.007
A2	0.770	0.820	0.870	0.030	0.032	0.034
A3	0.620	0.650	0.680	0.024	0.026	0.027
c	0.140	0.170	0.200	0.006	0.007	0.008
D	4.200	4.300	4.400	0.165	0.169	0.173
E	4.200	4.300	4.400	0.165	0.169	0.173
D1	-	3.600	-	-	0.142	-
E1	-	3.600	-	-	0.142	-
e	-	0.400	-	-	0.016	-
b	0.150	0.200	0.250	0.006	0.008	0.010
aaa	0.100			0.004		
bbb	0.100			0.004		
ddd	0.080			0.003		
eee	0.150			0.006		
fff	0.050			0.002		

Note:

Values in inches are converted from values in millimeter and rounded to 3 decimal digits.

10.2 QFN68

GR5526 QFN68, including GR5526RGNIP QFN68 and GR5526RGNI QFN68, is a 68-pin and 7 x 7 x 0.85 (mm) package. It is qualified to MSL 3.

Table 10-3 QFN68 package information

Parameter	Value	Unit	Tolerance
Package Size	7.0 x 7.0	mm	±0.05 mm
QFN Pad Count	68		
Total Thickness	0.85	mm	±0.05 mm
QFN Pad Pitch	0.35		

Parameter	Value	Unit	Tolerance
Pad Width	0.15		±0.05 mm
Exposed Pad Size	5.49 x 5.49		±0.1 mm

The figure below shows the QFN68 package outlines.

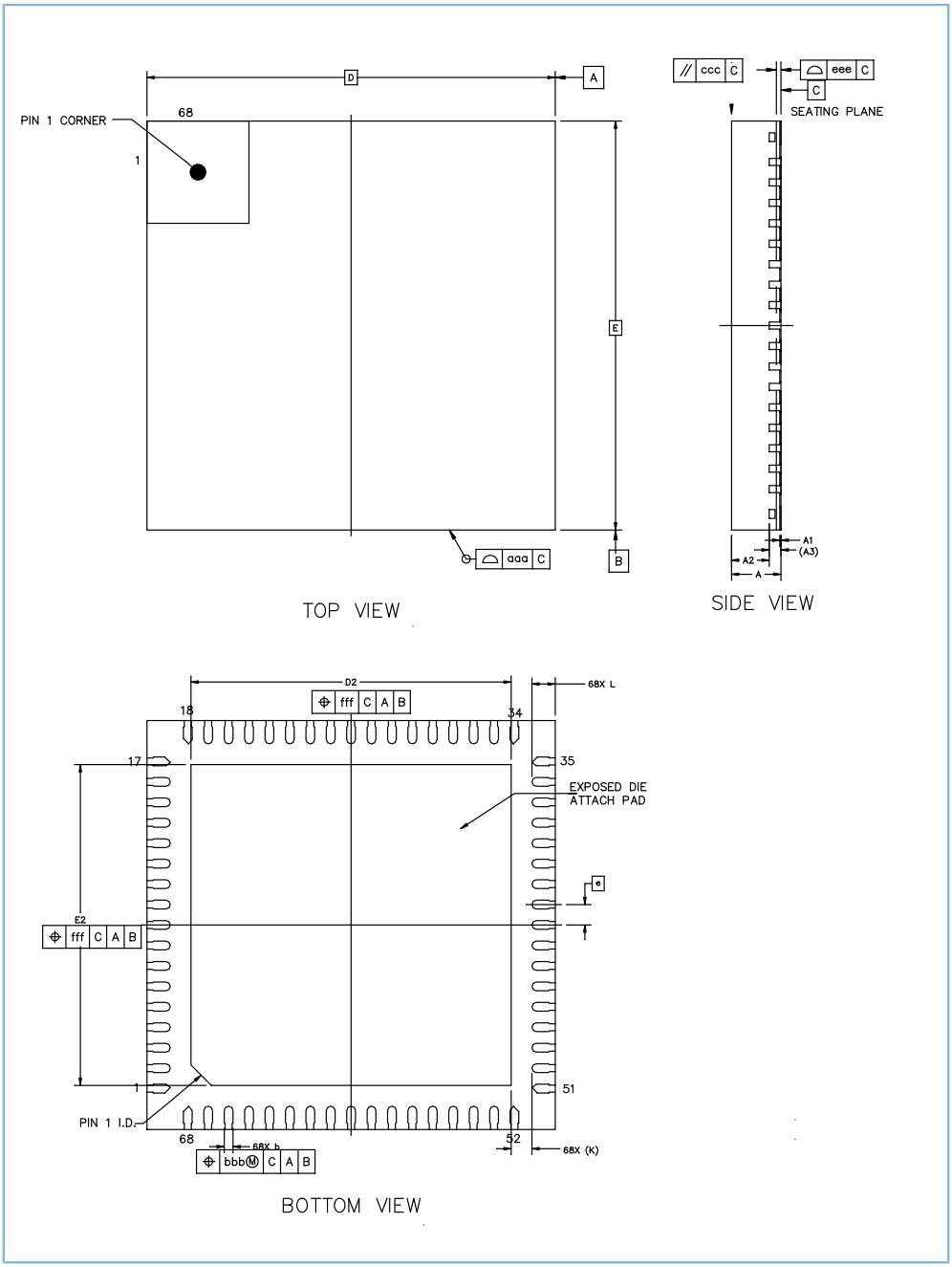

Figure 10-2 QFN68 package outlines

Note:

Drawing is not to scale.

Table 10-4 QFN68 package dimensions

Symbol	Dimensions in mm			Dimensions in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.800	0.850	0.900	0.031	0.033	0.035
A1	0.000	0.020	0.050	0.000	0.001	0.002
A2	-	0.650	-	-	0.026	-
A3	0.203 REF.			0.008 REF.		
b	0.100	0.150	0.200	0.004	0.006	0.008
D	7.000 BSC.			0.276 BSC.		
E	7.000 BSC.			0.276 BSC.		
e	0.350 BSC.			0.014 BSC.		
D2	5.390	5.490	5.590	0.212	0.216	0.220
E2	5.390	5.490	5.590	0.212	0.216	0.220
L	0.350	0.400	0.450	0.014	0.016	0.018
K	0.355 REF.			0.014 REF.		
aaa	0.100			0.004		
ccc	0.100			0.004		
eee	0.080			0.003		
bbb	0.070			0.003		
fff	0.100			0.004		

Note:

Values in inches are converted from values in millimeter and rounded to 3 decimal digits.

11 Ordering Codes

Taking GR5526VGBIP as an example, the figure bellow shows the ordering codes and definitions for GR5526 series.

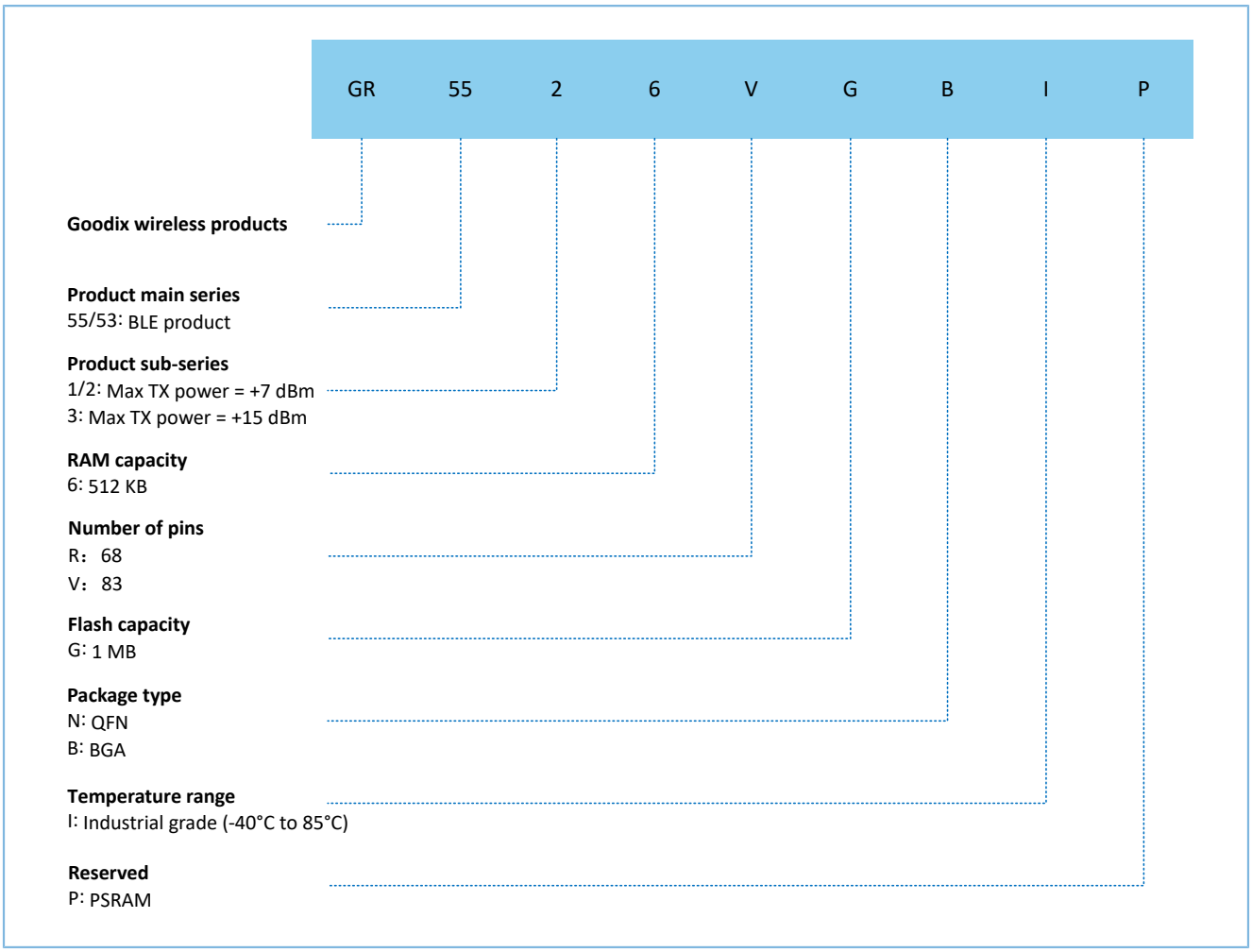

Figure 11-1 Ordering codes and definitions for GR5526 series

12 Tape and Reel Information

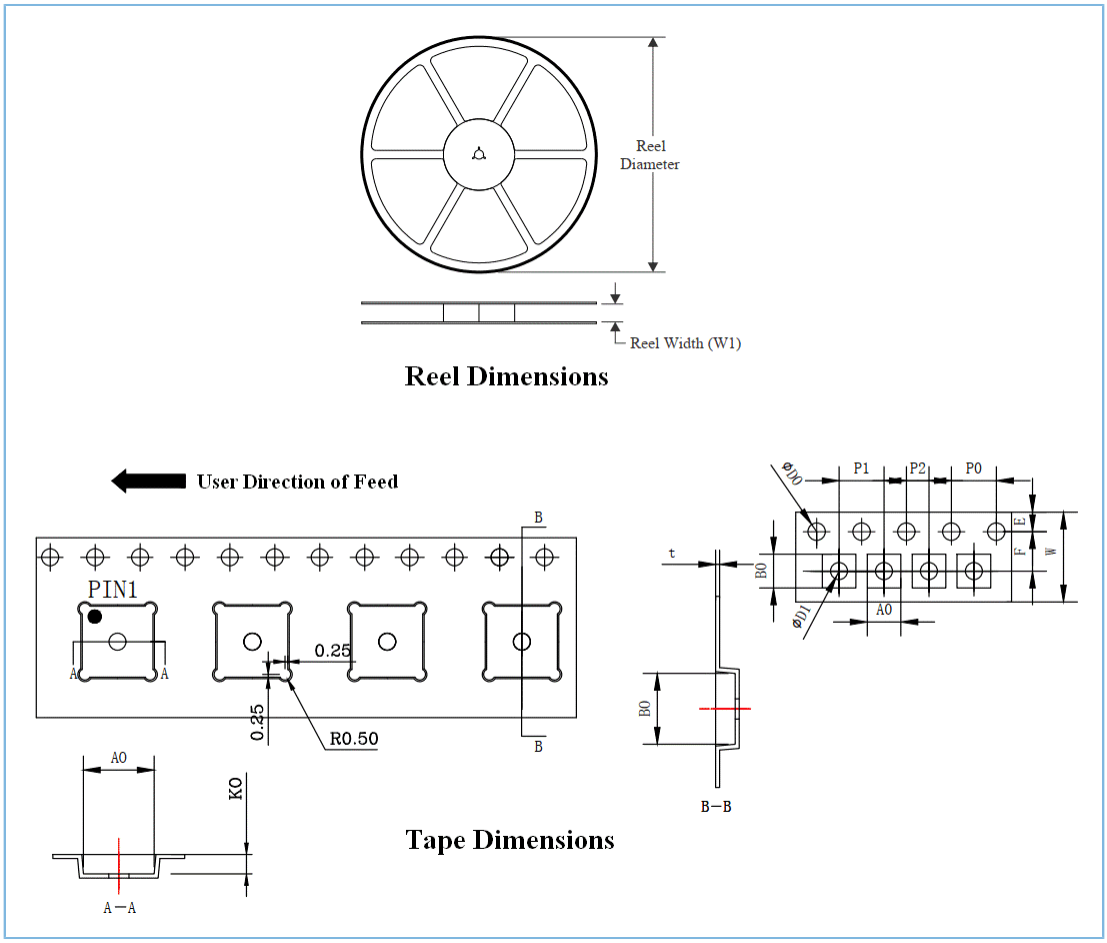

Figure 12-1 Tape and reel dimensions

Note:

- The design of carrier tape must meet EIA-481.
- Each 10 sprocket holes pitch cumulative tolerance is ± 0.2 mm.
- Carrier tape camber should not exceed 1 mm in 250 mm.
- Meet RoHs environment material management.

The detailed dimensions are outlined in the table below.

Table 12-1 Tape and reel dimensions (mm)

Package Type	Reel Diameter	Reel Width	W	E	F	D0	D1	P0	P2	10P0	P1	A0	B0	K0	t
QFN68 7 x 7	178.00	16.00	16.00 ±0.15	1.75 ±0.10	7.50 ±0.10	1.50 +0.10/-0	1.50 +0.10/-0	4.00 ±0.10	2.00 ±0.10	40.00 ±0.20	12.00 ±0.10	7.30 ±0.10	7.30 ±0.10	1.10 ±0.10	0.30 ±0.05

Package Type	Reel Diameter	Reel Width	W	E	F	D0	D1	P0	P2	10P0	P1	A0	B0	K0	t
BGA83 4.3 x 4.3	178.00	12.00	12.00 ±0.10	1.75 ±0.10	5.50 ±0.10	1.50 +0.10/-0	1.50 +0.10/-0	4.00 ±0.10	2.00 ±0.10	40.00 ±0.20	8.00 ±0.10	4.60 ±0.10	4.60 ±0.10	1.30 ±0.10	0.30 ±0.05

13 Glossary

Table 13-1 Glossary

Name	Description
ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AGC	Automatic Gain Control
AMBA	Advanced Microcontroller Bus Architecture
AMS	Analog Mix Signal
AoA/AoD	Angle of Arrival/Angle of Departure
AON	Always-on
AONWDG	Always-on Watchdog Timer
APB	Advanced Peripheral Bus
ATCSR	Auto Temperature Compensated Self-Refresh
BB	Baseband
Bluetooth LE	Bluetooth Low Energy
BOD	Brown-out Detector
CBC	Cipher Block Chaining
DC	Display Control
DFE	Digital Frontend
DPA	Differential Power Analysis
DSPI	Dual-lane SPI
DWT	Data Watchpoint Trigger
ECB	Electronic Codebook
ESL	Electronic Shelf Label
FIPS	Federal Information Processing Standards
FPB	Flash Patch and Breakpoint
FPU	Floating Point Unit
GPIO	General-Purpose I/O
HMAC	Hash-based Message Authentication Code
IFS	Inter Frame Spacing
IoT	Internet of Things
IRQ	Interrupt Request
ISRs	Interrupt Service Routines
ITM	Instrumentation Trace and Macrocell

Name	Description
IV	Initialization Vector
LDO	Low Dropout
LFU	Least Frequently Used
LNA	Low Noise Amplifier
LP Comp	Low-Power Comparator
LPD	Low Power Domain
LSB	Least Significant Bit
MPU	Memory Protection Unit
MSB	Most Significant Bit
MSIO	Mixed Signal I/O
NMI	Non-maskable Interrupt
NVIC	Nested Vectored Interrupt Controller
NVM	Non-volatile Memory
PA	Power Amplifier
PKC	Public Key Cryptography
PLL	Phase Locked Loop
PMU	Power Management Unit
POR	Power-on Reset
PSC	Power State Controller
QSPI	Quad SPI
RBX	Row boundary crossing
RNG	RING Oscillator
RTC	Real-Time Counters
SCL	Serial Clock Line
SDA	Serial Data Line
SHS	Secure Hash Standard
SiP	System-in-Package
SNR	Signal to Noise Ratio
SoC	System-on-Chip
SWD	Serial Wire Debug
SWO	Serial Wire Output
TPIU	Trace Port Interface Unit
TPMS	Tire Pressure Monitor System
TRNG	True Random Number Generator

Name	Description
Typ	Typical
ULP	Ultra Low Power
VIO	I/O Voltage
W1C	Write 1 to Clear
XO	Crystal Oscillator

14 Reference Documents

The following documents are referred in this datasheet:

Table 14-1 Reference documents

Documents	Description
GR5526 Hardware Design Guidelines	Introduce GR5526 hardware design.
GR5526 Developer User Guide	Introduce GR5526 SDK and its related resources to developers.

15 Legal and Contact Information

Copyright © 2024 Shenzhen Goodix Technology Co., Ltd. All rights reserved.

Any excerption, backup, modification, translation, transmission or commercial use of this document or any portion of this document, in any form or by any means, without the prior written consent of Shenzhen Goodix Technology Co., Ltd. is prohibited.

Trademarks and Permissions

GOODIX and other Goodix trademarks are trademarks of Shenzhen Goodix Technology Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Disclaimer

Information contained in this document is intended for your convenience only and is subject to change without prior notice. It is your responsibility to ensure its application complies with technical specifications.

Shenzhen Goodix Technology Co., Ltd. (hereafter referred to as “Goodix”) makes no representation or guarantee for this information, express or implied, oral or written, statutory or otherwise, including but not limited to representation or guarantee for its application, quality, performance, merchantability or fitness for a particular purpose. Goodix shall assume no responsibility for this information and relevant consequences arising out of the use of such information.

Without written consent of Goodix, it is prohibited to use Goodix products as critical components in any life support system. Under the protection of Goodix intellectual property rights, no license may be transferred implicitly or by any other means.

Shenzhen Goodix Technology Co., Ltd.

Headquarters: Floor 13, Phase B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China

TEL: +86-755-33338828 Zip Code: 518000

Website: www.goodix.com

16 Revision History

Table 16-1 Revision history

Version	Date	Description
1.0	2023-01-10	Initial release
1.1	2024-06-06	Added "Tape and Reel Information".
1.2	2024-09-27	Updated the section " Receiver Performance".