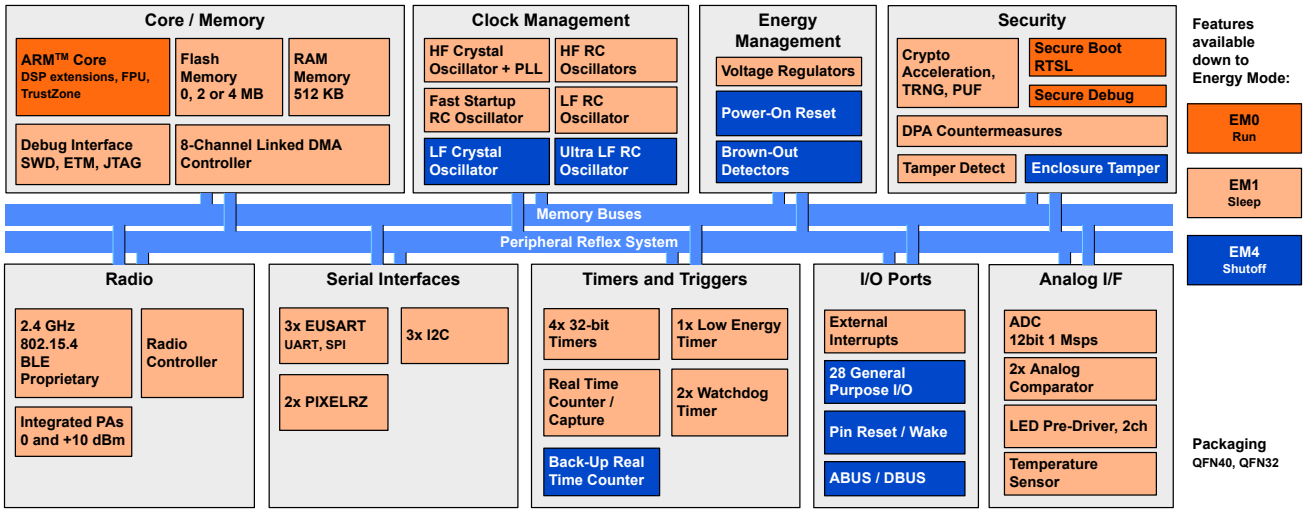
SiXG301 Wireless SoC Reference Manual

The SiXG301 Wireless SoCs are ideal for mesh IoT wireless connectivity using Matter, OpenThread, and Zigbee.

With key features like high-performance 2.4 GHz RF, low current consumption, and Secure Vault, IoT device makers can create smart, robust, and energy-efficient products that are secure from remote and local cyber-attacks. An ARM Cortex-M33® Core processor running up to 150 MHz with up to 4 MB of co-packaged flash and 512 KB of RAM provides resources for demanding applications while leaving room for future growth.

KEY FEATURES

- 32-bit ARM® core with 150 MHz maximum operating frequency
- Up to 4 MB of flash and 512 KB of RAM
- High-performance radio with up to +10 dBm output power
- Energy-efficient design with low active and sleep currents
- Secure Vault™



 Copyright © 2026 by Silicon Laboratories

Table of Contents

1. About This Document 27

1.1 Introduction 27

1.2 Conventions 28

1.3 Related Documentation 29

2. System Overview 30

2.1 Introduction 30

2.2 Block Diagrams 31

2.3 MCU Features overview 32

3. System Processor. 34

3.1 Introduction 34

3.2 Features 35

3.3 Functional Description 35

3.3.1 Interrupt Operation 36

3.3.2 TrustZone 36

3.3.3 Interrupt Request Lines (IRQ) 37

4. System Bus and Memory Map 40

4.1 Introduction 40

4.2 Functional Description 41

4.2.1 Bus Matrix 42

4.2.2 External Memory 43

4.2.3 RAM 43

4.2.4 Peripherals 43

5. System Memory and Configuration 49

5.1 Introduction 49

5.2 EXTMEM - External Memory Interface 49

5.2.1 Features 49

5.2.2 Overview. 50

5.2.3 Flash Partitioning 51

5.2.4 Write and Erase 52

5.2.5 Flash Regulator (FLREG) 52

5.2.6 Quad SPI (QSPI) 52

5.2.7 L2CACHE 53

5.3 HYDRARAM - RAM Controller 53

5.3.1 Features 53

5.3.2 RAM Bank Caching 53

5.3.3 RAM Bank Interleaving 54

5.3.4 Bus Address Faults 54

5.3.5 ECC 55

5.3.6 Port Arbitration Priority 55

5.3.7 HYDRARAM Register Map 56

5.3.8 HYDRARAM Register Description 58
5.4 ICACHE - Instruction Cache 66
5.4.1 Cache Operation 66
5.4.2 Performance Measurement 67
5.4.3 ICACHE Register Map 68
5.4.4 ICACHE Register Description 69
5.5 Device Information 74
5.5.1 Introduction 74
5.5.2 Description 74
5.5.3 DEVINFO Register Block 74
5.5.4 Device Info Fragments 75
5.6 SYSCFG - System Configuration 80
5.6.1 Software Interrupts 80
5.6.2 FPU Exceptions 80
5.6.3 Bus Faults 80
5.6.4 SYSCFG Register Map 81
5.6.5 SYSCFG Register Description 83
5.6.6 SYSCFG_CFGNS Register Map 91
5.6.7 SYSCFG_CFGNS Register Description 92
5.7 SEMAPHORE 93
5.7.1 Introduction 93
5.7.2 Description 93
6. Low-Power Wireless Transceiver (LPW) 95
6.1 Introduction 96
6.1.1 RF Frequency Synthesizer 96
6.1.2 Modulation Modes 97
6.1.3 Transmit Mode 97
6.1.4 Receive Mode 97
6.1.5 Data Buffering 97
6.1.6 Unbuffered Data Transfer 97
6.1.7 Frame Format Support 98
6.1.8 Hardware CRC Support 98
6.1.9 Convolutional Encoding / Decoding 98
6.1.10 Binary Block Encoding / Decoding 98
6.1.11 Data Encryption and Authentication 99
6.1.12 RF Test Modes 99
7. DBG - Debug Interface 100
7.1 Introduction 100
7.2 Features 100
7.3 Functional Description 101
7.3.1 Debug Pins 101
7.3.2 Embedded Trace Macrocell (ETM) 101
8. CMU - Clock Management Unit 102
8.1 Introduction 102

8.2 Features . . . 102
8.3 Functional Description . . . 103
8.3.1 System Clocks . . . 106
8.3.2 Switching Clock Source . . . 110
8.3.3 Oscillator Calibration and Measurement . . . 112
8.3.4 Energy Modes . . . 115
8.3.5 Clock Output . . . 116
8.3.6 Clock Input from a Pin . . . 116
8.3.7 Interrupts . . . 116
8.3.8 Protection . . . 116
8.4 CMU Register Map . . . 117
8.5 CMU Register Description . . . 120
8.5.1 CMU_IPVERSION - IP Version ID . . . 120
8.5.2 CMU_STATUS - Status Register . . . 121
8.5.3 CMU_LOCK - Configuration Lock Register . . . 122
8.5.4 CMU_WDOGLOCK - WDOG Configuration Lock Register . . . 122
8.5.5 CMU_IF - Interrupt Flag Register . . . 123
8.5.6 CMU_IEN - Interrupt Enable Register . . . 123
8.5.7 CMU_CALCMD - Calibration Command Register . . . 124
8.5.8 CMU_CALCTRL - Calibration Control Register . . . 125
8.5.9 CMU_CALCNT - Calibration Result Counter Register . . . 126
8.5.10 CMU_CALTOP - Calibration Top Value Register . . . 126
8.5.11 CMU_CLKEN0 - Clock Enable Register 0 . . . 127
8.5.12 CMU_CLKEN1 - Clock Enable Register 1 . . . 129
8.5.13 CMU_SYSCLKCTRL - System Clock Control . . . 131
8.5.14 CMU_TRACECLKCTRL - Debug Trace Clock Control . . . 132
8.5.15 CMU_EXPORTCLKCTRL - Export Clock Control . . . 133
8.5.16 CMU_DPLLREFCLKCTRL - Digital PLL Reference Clock Control . . . 135
8.5.17 CMU_EM01GRPACLKCTRL - EM01 Peripheral Group A Clock Control . . . 136
8.5.18 CMU_EM01GRPCCLKCTRL - EM01 Peripheral Group C Clock Control . . . 137
8.5.19 CMU_EM23GRPACLKCTRL - EM23 Peripheral Group A Clock Control . . . 138
8.5.20 CMU_EM4GRPACLKCTRL - EM4 Peripheral Group A Clock Control . . . 138
8.5.21 CMU_ADCCLKCTRL - ADC Clock Control . . . 139
8.5.22 CMU_WDOG0CLKCTRL - WDOG0 Clock Control . . . 139
8.5.23 CMU_EUSART0CLKCTRL - EUSART0 Clock Control . . . 140
8.5.24 CMU_SYSRTC0CLKCTRL - SYSRTC0 Clock Control . . . 140
8.5.25 CMU_PCNT0CLKCTRL - PCNT0 Clock Control . . . 141
8.5.26 CMU_RADIOCLKCTRL - Radio Clock Control . . . 141
8.5.27 CMU_HFXO0LFCLKCTRL - HFXO0 LF Clock Control . . . 142
8.5.28 CMU_EM01GRPDCLKCTRL - EM01 Peripheral Group D Clock Control . . . 142
8.5.29 CMU_I2C0CLKCTRL - I2C0 Clock Control . . . 143
8.5.30 CMU_PIXELRZCLKCTRL - PIXELRZ Clock Control . . . 144
8.5.31 CMU_CLKENHV - Clock Enable Register 2 . . . 145
9. Oscillators . . . 146
9.1 Introduction . . . 146
9.2 HFXO - High Frequency Crystal Oscillator . . . 146

9.2.1 Introduction . . . 146
9.2.2 Features . . . 147
9.2.3 Functional Description . . . 147
9.2.4 HFXO Register Map . . . 150
9.2.5 HFXO Register Description . . . 151
9.3 SOCPLL - SoC Phase-Locked-Loop . . . 168
9.3.1 Introduction . . . 168
9.3.2 Features . . . 169
9.3.3 Functional Description . . . 169
9.3.4 SOCPLL Register Map . . . 170
9.3.5 SOCPLL Register Description . . . 171
9.4 HFRCO - High-Frequency RC Oscillator . . . 175
9.4.1 Introduction . . . 175
9.4.2 Features . . . 175
9.4.3 Functional Description . . . 175
9.4.4 HFRCO Register Map . . . 178
9.4.5 HFRCO Register Description . . . 179
9.5 DPLL - Digital Phased Locked Loop . . . 182
9.5.1 Introduction . . . 182
9.5.2 Features . . . 183
9.5.3 Functional Description . . . 183
9.5.4 DPLL Register Map . . . 185
9.5.5 DPLL Register Description . . . 186
9.6 LFXO - Low-Frequency Crystal Oscillator . . . 190
9.6.1 Introduction . . . 190
9.6.2 Features . . . 190
9.6.3 Functional Description . . . 190
9.6.4 LFXO Register Map . . . 193
9.6.5 LFXO Register Description . . . 194
9.7 LFRCO - Low-Frequency RC Oscillator . . . 201
9.7.1 Introduction . . . 201
9.7.2 Features . . . 201
9.7.3 Functional Description . . . 201
9.7.4 LFRCO Register Map . . . 204
9.7.5 LFRCO Register Description . . . 205
9.8 FSRCO - Fast Start RCO . . . 211
9.8.1 Introduction . . . 211
9.8.2 Features . . . 211
9.8.3 Functional Description . . . 211
9.8.4 FSRCO Register Map . . . 211
9.8.5 FSRCO Register Description . . . 212
9.9 ULFRCO - Ultra Low Frequency RC Oscillator . . . 212
9.9.1 Introduction . . . 212
9.9.2 Features . . . 212
9.9.3 Functional Description . . . 212
10. SMU - Security Management Unit . . . 213

10.1 Introduction . . . 213
10.2 Features . . . 213
10.3 Functional Description . . . 214
10.3.1 Bus Level Security . . . 214
10.3.2 Secure and Non-Secure Register Blocks . . . 214
10.3.3 Privileged Access Control . . . 215
10.3.4 Secure Access Control . . . 215
10.3.5 ARM TrustZone . . . 216
10.3.6 Configuring Managers in the BMPU . . . 216
10.3.7 Configuring Peripherals in the PPU . . . 216
10.3.8 Configuring ESAU Memory Regions . . . 217
10.3.9 Configuring Memory in the MSPU . . . 218
10.3.10 Cortex-M33 Integration . . . 218
10.3.11 Exception Handling . . . 218
10.3.12 SMU Lock . . . 219
10.4 SMU Register Map . . . 220
10.5 SMU Register Description . . . 223
10.5.1 SMU_IPVERSION - IP Version . . . 223
10.5.2 SMU_STATUS - Status Register . . . 224
10.5.3 SMU_LOCK - Lock Register . . . 224
10.5.4 SMU_IF - Interrupt Flag Register . . . 225
10.5.5 SMU_IEN - Interrupt Enable Register . . . 226
10.5.6 SMU_M33CTRL - M33 Control Settings . . . 227
10.5.7 SMU_PPUPATD0 - PPU Privileged Access . . . 228
10.5.8 SMU_PPUPATD1 - PPU Privileged Access . . . 230
10.5.9 SMU_PPUSATD0 - PPU Secure Access . . . 232
10.5.10 SMU_PPUSATD1 - PPU Secure Access . . . 234
10.5.11 SMU_PPURSPERRD - PPU Respond With Error . . . 236
10.5.12 SMU_PPUFS - PPU Fault Status . . . 236
10.5.13 SMU_PPUFSADDR - PPU Fault Status Address . . . 237
10.5.14 SMU_BMPUPATD0 - BMPU Privileged Attribute . . . 238
10.5.15 SMU_BMPUSATD0 - BMPU Secure Attribute . . . 239
10.5.16 SMU_BMPURSPERRD0 - BMPU Respond With Error . . . 240
10.5.17 SMU_BMPUFS - BMPU Fault Status . . . 241
10.5.18 SMU_BMPUFSADDR - BMPU Fault Status Address . . . 241
10.5.19 SMU_MSPUDMEMREGIONSIZE - DMEM MSPU Region Size . . . 242
10.5.20 SMU_MSPUNVMREGIONSIZE - NVM MSPU Region Size . . . 243
10.5.21 SMU_MSPUDMEMRSPERRD - DMEM MSPU Respond With Error . . . 243
10.5.22 SMU_MSPUNVMRSPERRD - NVM MSPU Respond With Error . . . 244
10.5.23 SMU_MSPUDMEMNSREGIONFLAG0 - DMEM MSPU Non-Secure Region Flags . . . 245
10.5.24 SMU_MSPUNVMNSREGIONFLAG0 - NVM MSPU Non-Secure Region Flags . . . 247
10.5.25 SMU_MSPUNVMNSREGIONFLAG1 - NVM MSPU Non-Secure Region Flags . . . 249
10.5.26 SMU_MSPUNVMNSREGIONFLAG2 - NVM MSPU Non-Secure Region Flags . . . 251
10.5.27 SMU_MSPUNVMNSREGIONFLAG3 - NVM MSPU Non-Secure Region Flags . . . 253
10.5.28 SMU_MSPUDMEMFSADDR - DMEM MSPU Fault Status Address . . . 255
10.5.29 SMU_MSPUNVMFSADDR - NVM MSPU Fault Status Address . . . 255
10.5.30 SMU_ESAURTYPES - ESAU Region Types . . . 256

10.6 SMU_CFGNS Register Map . . . 257
10.7 SMU_CFGNS Register Description . . . 258
10.7.1 SMU_NSSTATUS - Status Register . . . 258
10.7.2 SMU_NSLOCK - Lock Register . . . 258
10.7.3 SMU_NSIF - Interrupt Flag Register . . . 259
10.7.4 SMU_NSIEN - Interrupt Enable Register . . . 259
10.7.5 SMU_PPUNSPATD0 - PPU Privileged Access . . . 260
10.7.6 SMU_PPUNSPATD1 - PPU Privileged Access . . . 262
10.7.7 SMU_PPUNSFS - PPU Fault Status . . . 264
10.7.8 SMU_PPUNSFSADDR - PPU Fault Status Address . . . 264
10.7.9 SMU_BMPUNSPATD0 - BMPU Privileged Attribute . . . 265
11. SYMCRYPTO - Symmetric Cryptography Engine . . . 266
11.1 Introduction . . . 266
11.2 Features . . . 266
11.3 Cryptographic Functions . . . 266
11.3.1 SYMCRYPTO Register Map . . . 267
11.3.2 SYMCRYPTO Register Description . . . 268
12. ETAMPDET - External Tamper Detect . . . 286
12.1 Introduction . . . 286
12.2 Features . . . 286
12.3 Functional Description . . . 287
12.3.1 Configuration . . . 287
12.3.2 Clocking . . . 287
12.3.3 Filtering . . . 288
12.3.4 Lock / Unlock . . . 288
12.3.5 Wakeup and Interrupt Sources . . . 288
12.3.6 Commands . . . 289
12.4 ETAMPDET Register Map . . . 290
12.5 ETAMPDET Register Description . . . 292
12.5.1 ETAMPDET_IPVERSION - IP Version ID . . . 292
12.5.2 ETAMPDET_EN - Module Enable Register . . . 292
12.5.3 ETAMPDET_CFG - Configuration Register . . . 293
12.5.4 ETAMPDET_CNTMISMATCHMAX - Filter Threshold Register . . . 295
12.5.5 ETAMPDET_CHNLFILTWINSIZE - Filter Moving Window Size Register . . . 297
12.5.6 ETAMPDET_CMD - Command Register . . . 299
12.5.7 ETAMPDET_SYNCBUSY - Syncbusy Status Register . . . 300
12.5.8 ETAMPDET_IEN - Interrupt Enable Register . . . 300
12.5.9 ETAMPDET_IF - Interrupt Flag Register . . . 301
12.5.10 ETAMPDET_STATUS - Status Register . . . 301
12.5.11 ETAMPDET_EM4WUEN - EM4 Wakeup Request Enable Register . . . 302
12.5.12 ETAMPDET_CHNLSEEDVAL0 - CHNL0 LFSR Seed Ctrl Register . . . 302
12.5.13 ETAMPDET_CHNLSEEDVAL1 - CHNL1 LFSR Seed Ctrl Register . . . 303
12.5.14 ETAMPDET_CLKPRESCVAL - Prescaler Ctrl Register . . . 304
12.5.15 ETAMPDET_LOCK - Configuration Lock Register . . . 306

13. EMU - Energy Management Unit . . . 307
13.1 Introduction . . . 307
13.2 Features . . . 307
13.3 Functional Description . . . 308
13.3.1 Energy Modes . . . 309
13.3.2 EM0 . . . 309
13.3.3 EM1 . . . 309
13.3.4 EM4 . . . 310
13.4 Entering Low Energy Modes . . . 310
13.4.1 Entry Into EM1 . . . 310
13.4.2 Entry Into EM4 . . . 310
13.5 Exiting Low Energy Modes . . . 310
13.5.1 Exiting From EM1 . . . 310
13.5.2 Exiting From EM4 . . . 310
13.6 Power Configurations . . . 310
13.6.1 Power Configuration 1: Single 3.3 V Supply . . . 311
13.6.2 Power Configuration 2: Single 1.9 - 3.6 V Supply . . . 312
13.6.3 Power Configuration 3: Single 1.8 V Supply . . . 313
13.6.4 Other Supply Configurations . . . 313
13.7 Power Domains . . . 314
13.8 Brown Out Detector (BOD) . . . 314
13.9 Reset Management Unit . . . 315
13.10 Temperature Sensor . . . 317
13.10.1 Register Locks . . . 317
13.11 EMU Register Map . . . 318
13.12 EMU Register Description . . . 320
13.12.1 EMU_DECBOD - DECOUPLE LVBOD Control Register . . . 320
13.12.2 EMU_BOD3SENSE - BOD3SENSE Control Register . . . 321
13.12.3 EMU_FLREG - FLREG Control Register . . . 322
13.12.4 EMU_FLBOD - FLBOD Control Register . . . 322
13.12.5 EMU_IPVERSION - IP Version . . . 323
13.12.6 EMU_LOCK - EMU Configuration Lock Register . . . 323
13.12.7 EMU_IF - Interrupt Flags . . . 324
13.12.8 EMU_IEN - Interrupt Enables . . . 325
13.12.9 EMU_EM4CTRL - EM4 Control . . . 326
13.12.10 EMU_CMD - EMU Command Register . . . 327
13.12.11 EMU_CTRL - EMU Control Register . . . 328
13.12.12 EMU_TEMPLIMITS - EMU Temperature Thresholds . . . 329
13.12.13 EMU_STATUS - EMU Status Register . . . 330
13.12.14 EMU_TEMP - Temperature . . . 331
13.12.15 EMU_RSTCTRL - Reset Management Control Register . . . 332
13.12.16 EMU_RSTCAUSE - Reset Cause . . . 334
14. PRS - Peripheral Reflex System . . . 335
14.1 Introduction . . . 335

14.2 Features . . . 335
14.3 Functional Description . . . 336
14.3.1 Asynchronous Channel Functions . . . 336
14.3.2 Configurable Logic . . . 337
14.3.3 Producers . . . 338
14.3.4 Consumers . . . 345
14.4 PRS Register Map . . . 347
14.5 PRS Register Description . . . 362
14.5.1 PRS_IPVERSION - PRS IPVERSION . . . 362
14.5.2 PRS_ASYNC_SWPULSE - Software Pulse Register . . . 363
14.5.3 PRS_ASYNC_SWLEVEL - Software Level Register . . . 364
14.5.4 PRS_ASYNC_PEEK - Async Channel Values . . . 365
14.5.5 PRS_SYNC_PEEK - Sync Channel Values . . . 366
14.5.6 PRS_ASYNC_CHx_CTRL - Async Channel Control Register . . . 367
14.5.7 PRS_SYNC_CHx_CTRL - Sync Channel Control Register . . . 368
14.5.8 PRS_CONSUMER_EUSART0_CLK - CLK Consumer Register . . . 368
14.5.9 PRS_CONSUMER_EUSART0_RX - RX Consumer Register . . . 369
14.5.10 PRS_CONSUMER_EUSART0_TRIGGER - TRIGGER Consumer Register . . . 369
14.5.11 PRS_CONSUMER_EUSART1_CLK - CLK Consumer Register . . . 370
14.5.12 PRS_CONSUMER_EUSART1_RX - RX Consumer Register . . . 370
14.5.13 PRS_CONSUMER_EUSART1_TRIGGER - TRIGGER Consumer Register . . . 371
14.5.14 PRS_CONSUMER_EUSART2_CLK - CLK Consumer Register . . . 371
14.5.15 PRS_CONSUMER_EUSART2_RX - RX Consumer Register . . . 372
14.5.16 PRS_CONSUMER_EUSART2_TRIGGER - TRIGGER Consumer Register . . . 372
14.5.17 PRS_CONSUMER_FRC_RXRAW - RXRAW Consumer Register . . . 373
14.5.18 PRS_CONSUMER_CMU_CALDN - CALDN Consumer Register . . . 373
14.5.19 PRS_CONSUMER_CMU_CALUP - CALUP Consumer Register . . . 374
14.5.20 PRS_CONSUMER_LDMAXBAR0_DMAREQ0 - DMAREQ0 Consumer Register . . . 374
14.5.21 PRS_CONSUMER_LDMAXBAR0_DMAREQ1 - DMAREQ1 Consumer Register . . . 375
14.5.22 PRS_CONSUMER_LEDDRV0_LEDDRV0ASYNC - LEDDRV0 Consumer Register . . . 375
14.5.23 PRS_CONSUMER_LEDDRV0_LEDDRV1ASYNC - LEDDRV1 Consumer Register . . . 376
14.5.24 PRS_CONSUMER_LETIMER0_CLEAR - CLEAR Consumer Register . . . 376
14.5.25 PRS_CONSUMER_LETIMER0_START - START Consumer Register . . . 377
14.5.26 PRS_CONSUMER_LETIMER0_STOP - STOP Consumer Register . . . 377
14.5.27 PRS_CONSUMER_MODEM_DIN - DIN Consumer Register . . . 378
14.5.28 PRS_CONSUMER_MODEM_PAEN - PAEN Consumer Register . . . 378
14.5.29 PRS_CONSUMER_PCNT0_S0IN - S0IN Consumer Register . . . 379
14.5.30 PRS_CONSUMER_PCNT0_S1IN - S1IN Consumer Register . . . 379
14.5.31 PRS_CONSUMER_PIXELRZ0_INASYNC - IN Consumer Register . . . 380
14.5.32 PRS_CONSUMER_PIXELRZ1_INASYNC - IN Consumer Register . . . 380
14.5.33 PRS_CONSUMER_RAC_CLR - CLR Consumer Register . . . 381
14.5.34 PRS_CONSUMER_RAC_CTIIN0 - CTI Consumer Register . . . 381
14.5.35 PRS_CONSUMER_RAC_CTIIN1 - CTI Consumer Register . . . 382
14.5.36 PRS_CONSUMER_RAC_CTIIN2 - CTI Consumer Register . . . 382
14.5.37 PRS_CONSUMER_RAC_CTIIN3 - CTI Consumer Register . . . 383
14.5.38 PRS_CONSUMER_RAC_FORCETX - FORCETX Consumer Register . . . 383
14.5.39 PRS_CONSUMER_RAC_RXDIS - RXDIS Consumer Register . . . 384

14.5.40 PRS_CONSUMER_RAC_RXEN - RXEN Consumer Register 384
14.5.41 PRS_CONSUMER_RAC_TXEN - TXEN Consumer Register 385
14.5.42 PRS_CONSUMER_RFTIMER_PULSE - PULSE Consumer Register 385
14.5.43 PRS_CONSUMER_ADC0_ASYNCSEQ - ASYNC Consumer Register 386
14.5.44 PRS_CONSUMER_ADC0_SYNCSEQ - SYNC Consumer Register 386
14.5.45 PRS_CONSUMER_SEQACC_WAIT0 - WAIT0 Consumer Register 387
14.5.46 PRS_CONSUMER_SEQACC_WAIT1 - WAIT1 Consumer Register 387
14.5.47 PRS_CONSUMER_SEQACC_WAIT2 - WAIT2 Consumer Register 388
14.5.48 PRS_CONSUMER_SEQACC_WAIT3 - WAIT3 Consumer Register 388
14.5.49 PRS_CONSUMER_SETAMPER_TAMPERSRC29 - TAMPERSRC29 Consumer Register 389
14.5.50 PRS_CONSUMER_SETAMPER_TAMPERSRC30 - TAMPERSRC30 Consumer Register 389
14.5.51 PRS_CONSUMER_SETAMPER_TAMPERSRC31 - TAMPERSRC31 Consumer Register 390
14.5.52 PRS_CONSUMER_SYSRTC0_IN0 - IN0 Consumer Register 390
14.5.53 PRS_CONSUMER_SYSRTC0_IN1 - IN1 Consumer Register 391
14.5.54 PRS_CONSUMER_HFXO0_OSCREQ - OSC Consumer Register 391
14.5.55 PRS_CONSUMER_HFXO0_TIMEOUT - TIMEOUT Consumer Register 392
14.5.56 PRS_CONSUMER_CORE_CTIIN0 - CTI Consumer Register 392
14.5.57 PRS_CONSUMER_CORE_CTIIN1 - CTI Consumer Register 393
14.5.58 PRS_CONSUMER_CORE_CTIIN2 - CTI Consumer Register 393
14.5.59 PRS_CONSUMER_CORE_CTIIN3 - CTI Consumer Register 394
14.5.60 PRS_CONSUMER_CORE_M33RXEV - M33 Consumer Register. 394
14.5.61 PRS_CONSUMER_TIMER0_CC0 - CC0 Consumer Register 395
14.5.62 PRS_CONSUMER_TIMER0_CC1 - CC1 Consumer Register 395
14.5.63 PRS_CONSUMER_TIMER0_CC2 - CC2 Consumer Register 396
14.5.64 PRS_CONSUMER_TIMER0_DTI - DTI Consumer Register. 396
14.5.65 PRS_CONSUMER_TIMER0_DTIFS1 - DTI Consumer Register 397
14.5.66 PRS_CONSUMER_TIMER0_DTIFS2 - DTI Consumer Register 397
14.5.67 PRS_CONSUMER_TIMER1_CC0 - CC0 Consumer Register 398
14.5.68 PRS_CONSUMER_TIMER1_CC1 - CC1 Consumer Register 398
14.5.69 PRS_CONSUMER_TIMER1_CC2 - CC2 Consumer Register 399
14.5.70 PRS_CONSUMER_TIMER1_DTI - DTI Consumer Register. 399
14.5.71 PRS_CONSUMER_TIMER1_DTIFS1 - DTI Consumer Register 400
14.5.72 PRS_CONSUMER_TIMER1_DTIFS2 - DTI Consumer Register 400
14.5.73 PRS_CONSUMER_TIMER2_CC0 - CC0 Consumer Register 401
14.5.74 PRS_CONSUMER_TIMER2_CC1 - CC1 Consumer Register 401
14.5.75 PRS_CONSUMER_TIMER2_CC2 - CC2 Consumer Register 402
14.5.76 PRS_CONSUMER_TIMER2_CC3 - CC3 Consumer Register 402
14.5.77 PRS_CONSUMER_TIMER2_CC4 - CC4 Consumer Register 403
14.5.78 PRS_CONSUMER_TIMER2_CC5 - CC5 Consumer Register 403
14.5.79 PRS_CONSUMER_TIMER2_CC6 - CC6 Consumer Register 404
14.5.80 PRS_CONSUMER_TIMER2_DTI - DTI Consumer Register. 404
14.5.81 PRS_CONSUMER_TIMER2_DTIFS1 - DTI Consumer Register 405
14.5.82 PRS_CONSUMER_TIMER2_DTIFS2 - DTI Consumer Register 405
14.5.83 PRS_CONSUMER_TIMER3_CC0 - CC0 Consumer Register 406
14.5.84 PRS_CONSUMER_TIMER3_CC1 - CC1 Consumer Register 406
14.5.85 PRS_CONSUMER_TIMER3_CC2 - CC2 Consumer Register 407
14.5.86 PRS_CONSUMER_TIMER3_CC3 - CC3 Consumer Register 407
14.5.87 PRS_CONSUMER_TIMER3_CC4 - CC4 Consumer Register 408

14.5.88 PRS_CONSUMER_TIMER3_CC5 - CC5 Consumer Register . . . 408
14.5.89 PRS_CONSUMER_TIMER3_CC6 - CC6 Consumer Register . . . 409
14.5.90 PRS_CONSUMER_TIMER3_DTI - DTI Consumer Register . . . 409
14.5.91 PRS_CONSUMER_TIMER3_DTIFS1 - DTI Consumer Register . . . 410
14.5.92 PRS_CONSUMER_TIMER3_DTIFS2 - DTI Consumer Register . . . 410
14.5.93 PRS_CONSUMER_WDOG0_SRC0 - SRC0 Consumer Register . . . 411
14.5.94 PRS_CONSUMER_WDOG0_SRC1 - SRC1 Consumer Register . . . 411
14.5.95 PRS_CONSUMER_WDOG1_SRC0 - SRC0 Consumer Register . . . 412
14.5.96 PRS_CONSUMER_WDOG1_SRC1 - SRC1 Consumer Register . . . 412
14.5.97 PRS_RPURATD4 - Root Access Type Descriptor Register . . . 413

15. GPCRC - General Purpose Cyclic Redundancy Check . . . 414

15.1 Introduction . . . 414
15.2 Features . . . 414
15.3 Functional Description . . . 415
15.3.1 Polynomial Specification . . . 416
15.3.2 Input and Output Specification . . . 416
15.3.3 Initialization . . . 416
15.3.4 DMA Usage . . . 416
15.3.5 Byte-Level Bit Reversal and Byte Reordering . . . 417
15.4 GPCRC Register Map . . . 419
15.5 GPCRC Register Description . . . 420
15.5.1 GPCRC_IPVERSION - IPVERSION . . . 420
15.5.2 GPCRC_EN - CRC Enable . . . 420
15.5.3 GPCRC_CTRL - Control Register . . . 421
15.5.4 GPCRC_CMD - Command Register . . . 422
15.5.5 GPCRC_INIT - CRC Init Value . . . 422
15.5.6 GPCRC_POLY - CRC Polynomial Value . . . 422
15.5.7 GPCRC_INPUTDATA - Input 32-Bit Data Register . . . 423
15.5.8 GPCRC_INPUTDATAHWORD - Input 16-Bit Data Register . . . 423
15.5.9 GPCRC_INPUTDATABYTE - Input 8-Bit Data Register . . . 424
15.5.10 GPCRC_DATA - CRC Data Register . . . 424
15.5.11 GPCRC_DATAREV - CRC Data Reverse Register . . . 425
15.5.12 GPCRC_DATABYTEREV - CRC Data Byte Reverse Register . . . 425

16. SYSRTC - System RTC . . . 426

16.1 Introduction . . . 426
16.2 Features . . . 426
16.3 Functional Description . . . 426
16.3.1 Interrupts and Wake Events . . . 427
16.3.2 Counter . . . 427
16.3.3 Compare Events . . . 427
16.3.4 Capture Events . . . 427
16.3.5 SYSRTC Behavior on SWRST/Disablement/STOP . . . 427
16.3.6 Debug Functionality . . . 428
16.4 SYSRTC Register Map . . . 429

16.5 SYSRTC Register Description . . . 431
16.5.1 SYSRTC_IPVERSION - IP VERSION . . . 431
16.5.2 SYSRTC_EN - Module Enable Register . . . 432
16.5.3 SYSRTC_SWRST - Software Reset Register . . . 432
16.5.4 SYSRTC_CFG - Configuration Register . . . 433
16.5.5 SYSRTC_CMD - Command Register . . . 433
16.5.6 SYSRTC_STATUS - Status Register . . . 434
16.5.7 SYSRTC_CNT - Counter Value Register . . . 434
16.5.8 SYSRTC_SYNCBUSY - Synchronization Busy Register . . . 435
16.5.9 SYSRTC_LOCK - Configuration Lock Register . . . 435
16.5.10 SYSRTC_MSCNT - Millisecond Counter Register . . . 436
16.5.11 SYSRTC_MSCMPVAL - Millisecond Compare Register . . . 436
16.5.12 SYSRTC_MSCMPBUF - Millisecond Compare Buffer Register . . . 436
16.5.13 SYSRTC_IF - Interrupt Flag Register . . . 437
16.5.14 SYSRTC_IEN - Interrupt Enable Register . . . 437
16.5.15 SYSRTC_GRP0_IF - Group Interrupt Flags . . . 438
16.5.16 SYSRTC_GRP0_IEN - Group Interrupt Enables . . . 438
16.5.17 SYSRTC_GRP0_CTRL - Group Control Register . . . 439
16.5.18 SYSRTC_GRP0_CMP0VALUE - Compare 0 Value Register . . . 440
16.5.19 SYSRTC_GRP0_CMP1VALUE - Compare 1 Value Register . . . 440
16.5.20 SYSRTC_GRP0_CAP0VALUE - Capture 0 Value Register . . . 441
16.5.21 SYSRTC_GRP0_SYNCBUSY - Synchronization Busy Register . . . 441
16.5.22 SYSRTC_GRP0_PRETRIG - Pretrigger Register . . . 442

17. BURTC - Back-Up Real Time Counter . . . 443

17.1 Introduction . . . 443
17.2 Features . . . 443
17.3 Functional Description . . . 444
17.3.1 Clock Selection . . . 444
17.3.2 Configuration . . . 444
17.3.3 Debug Features and Description . . . 444
17.3.4 Counter . . . 445
17.3.5 Compare Channel . . . 445
17.3.6 Interrupts . . . 446
17.3.7 Register Lock . . . 446
17.4 BURTC Register Map . . . 447
17.5 BURTC Register Description . . . 448
17.5.1 BURTC_IPVERSION - IP Version ID . . . 448
17.5.2 BURTC_EN - Module Enable Register . . . 449
17.5.3 BURTC_CFG - Configuration Register . . . 450
17.5.4 BURTC_CMD - Command Register . . . 451
17.5.5 BURTC_STATUS - Status Register . . . 451
17.5.6 BURTC_IF - Interrupt Flag Register . . . 452
17.5.7 BURTC_IEN - Interrupt Enable Register . . . 452
17.5.8 BURTC_PRECNT - Pre-Counter Value Register . . . 453
17.5.9 BURTC_CNT - Counter Value Register . . . 453
17.5.10 BURTC_EM4WUEN - EM4 Wakeup Request Enable Register . . . 454

17.5.11 BURTC_SYNCBUSY - Synchronization Busy Register . . . 454
17.5.12 BURTC_LOCK - Configuration Lock Register . . . 455
17.5.13 BURTC_COMP - Compare Value Register . . . 455

18. BURAM - Backup RAM . . . 456

18.1 Introduction . . . 456
18.2 Functional Description . . . 456
18.3 BURAM Register Map . . . 456
18.4 BURAM Register Description . . . 457
18.4.1 BURAM_RETx_REG - Retention Register . . . 457

19. LETIMER - Low Energy Timer . . . 458

19.1 Introduction . . . 458
19.2 Features . . . 458
19.3 Functional Description . . . 459
19.3.1 Internal Overview . . . 460
19.3.2 Free Running Mode . . . 461
19.3.3 One-shot Mode . . . 462
19.3.4 Buffered Mode . . . 463
19.3.5 Double Mode . . . 464
19.4 Clock Frequency . . . 465
19.5 PRS Input Triggers . . . 466
19.6 Debug . . . 466
19.7 Output Action . . . 467
19.8 PRS Output . . . 467
19.9 Interrupts . . . 467
19.10 Register Access . . . 467
19.11 Programmer's Model . . . 468
19.11.1 Free Running Mode . . . 468
19.11.2 One Shot Mode . . . 469
19.11.3 DOUBLE Mode . . . 469
19.11.4 BUFFERED Mode . . . 470
19.11.5 Continuous Output Generation . . . 471
19.11.6 PWM Output . . . 472
19.12 LETIMER Register Map . . . 473
19.13 LETIMER Register Description . . . 475
19.13.1 LETIMER_IPVERSION - IP Version . . . 475
19.13.2 LETIMER_EN - Module Enable Register . . . 475
19.13.3 LETIMER_SWRST - Software Reset Register . . . 476
19.13.4 LETIMER_CTRL - Control Register . . . 477
19.13.5 LETIMER_CMD - Command Register . . . 479
19.13.6 LETIMER_STATUS - Status Register . . . 480
19.13.7 LETIMER_CNT - Counter Value Register . . . 480
19.13.8 LETIMER_COMP0 - Compare Value Register 0 . . . 481

19.13.9 LETIMER_COMP1 - Compare Value Register 1 . . . 481
19.13.10 LETIMER_TOP - Counter TOP Value Register . . . 481
19.13.11 LETIMER_TOPBUFF - Buffered Counter TOP Value . . . 482
19.13.12 LETIMER_REP0 - Repeat Counter Register 0 . . . 482
19.13.13 LETIMER_REP1 - Repeat Counter Register 1 . . . 482
19.13.14 LETIMER_IF - Interrupt Flag Register . . . 483
19.13.15 LETIMER_IEN - Interrupt Enable Register . . . 484
19.13.16 LETIMER_LOCK - Configuration Lock Register . . . 484
19.13.17 LETIMER_SYNCBUSY - Synchronization Busy Register . . . 485
19.13.18 LETIMER_PRSMODE - PRS Input Mode Select Register . . . 486

20. TIMER - Timer/Counter . . . 488

20.1 Introduction . . . 488

20.2 Features . . . 489

20.3 Functional Description . . . 490
20.3.1 Register Access . . . 490
20.3.2 Counter Modes . . . 491
20.3.3 Compare/Capture Channels . . . 497
20.3.4 Dead-Time Insertion Unit . . . 510
20.3.5 Debug Mode . . . 514
20.3.6 Interrupts, DMA and PRS Output . . . 515
20.3.7 GPIO Input/Output . . . 515

20.4 TIMER Register Map . . . 516

20.5 TIMER Register Description . . . 519
20.5.1 TIMER_IPVERSION - IP Version ID . . . 519
20.5.2 TIMER_CFG - Configuration Register . . . 520
20.5.3 TIMER_CTRL - Control Register . . . 523
20.5.4 TIMER_CMD - Command Register . . . 524
20.5.5 TIMER_STATUS - Status Register . . . 525
20.5.6 TIMER_STATUS2 - Status Reg for Phase and Dither . . . 527
20.5.7 TIMER_IF - Interrupt Flag Register . . . 528
20.5.8 TIMER_IEN - Interrupt Enable Register . . . 530
20.5.9 TIMER_TOP - Counter Top Value Register . . . 531
20.5.10 TIMER_TOPB - Counter Top Value Buffer Register . . . 531
20.5.11 TIMER_CNT - Counter Value Register . . . 532
20.5.12 TIMER_LOCK - TIMER Configuration Lock Register . . . 532
20.5.13 TIMER_EN - Module En . . . 533
20.5.14 TIMER_CCx_CFG - CC Channel Configuration Register . . . 534
20.5.15 TIMER_CCx_CTRL - CC Channel Control Register . . . 536
20.5.16 TIMER_CCx_PHASE - Channel Phase Register . . . 537
20.5.17 TIMER_CCx_PHASEB - Channel Phase Buffer Register . . . 538
20.5.18 TIMER_CCx_OC - OC Channel Value Register . . . 538
20.5.19 TIMER_CCx_OCB - OC Channel Value Buffer Register . . . 538
20.5.20 TIMER_CCx_DITHER - Dither Value Register . . . 539
20.5.21 TIMER_CCx_DITHERB - Dither Value Buffer Register . . . 539
20.5.22 TIMER_CCx_ICF - IC Channel Value Register . . . 539
20.5.23 TIMER_CCx_ICOF - IC Channel Value Overflow Register . . . 540

20.5.24 TIMER_DTCFG - DTI Configuration Register . . . 540
20.5.25 TIMER_DTTIMECFG - DTI Time Configuration Register . . . 541
20.5.26 TIMER_DTFCFG - DTI Fault Configuration Register . . . 542
20.5.27 TIMER_DTCTRL - DTI Control Register . . . 543
20.5.28 TIMER_DTOGEN - DTI Output Generation Enable Register . . . 543
20.5.29 TIMER_DTFAULT - DTI Fault Register . . . 544
20.5.30 TIMER_DTFAULTC - DTI Fault Clear Register . . . 545
20.5.31 TIMER_DTLOCK - DTI Configuration Lock Register . . . 545

21. EUSART - Universal Synchronous Asynchronous Receiver/Transmitter . . . 546

21.1 Introduction . . . 546

21.2 Features . . . 547

21.3 Functional Description . . . 548
21.3.1 Power Domains and Energy Modes . . . 548
21.3.2 Modes of Operation . . . 549
21.3.3 Asynchronous Operation . . . 549
21.3.4 Synchronous Operation . . . 571
21.3.5 Debug Halt . . . 574
21.3.6 PRS-triggered Transmissions . . . 574
21.3.7 PRS RX Input . . . 574
21.3.8 PRS CLK Input . . . 575
21.3.9 DMA Support . . . 575

21.4 EUSART Register Map . . . 576

21.5 EUSART Register Description . . . 578
21.5.1 EUSART_IPVERSION - IP Version ID . . . 578
21.5.2 EUSART_EN - Enable Register . . . 579
21.5.3 EUSART_CFG0 - Configuration 0 Register . . . 580
21.5.4 EUSART_CFG1 - Configuration 1 Register . . . 583
21.5.5 EUSART_CFG2 - Configuration 2 Register . . . 589
21.5.6 EUSART_FRAMECFG - Frame Format Register . . . 591
21.5.7 EUSART_DTXDATCFG - Default TX DATA Register . . . 592
21.5.8 EUSART_IRHFCFG - HF IrDA Mod Config Register . . . 593
21.5.9 EUSART_IRLFCFG - LF IrDA Pulse Config Register . . . 594
21.5.10 EUSART_TIMINGCFG - Timing Register . . . 595
21.5.11 EUSART_STARTFRAMECFG - Start Frame Register . . . 597
21.5.12 EUSART_SIGFRAMECFG - Signal Frame Register . . . 597
21.5.13 EUSART_CLKDIV - Clock Divider Register . . . 598
21.5.14 EUSART_TRIGCTRL - Trigger Control Register . . . 598
21.5.15 EUSART_CMD - Command Register . . . 599
21.5.16 EUSART_RXDATA - RX Data Register . . . 600
21.5.17 EUSART_RXDATAP - RX Data Peek Register . . . 600
21.5.18 EUSART_TXDATA - TX Data Register . . . 601
21.5.19 EUSART_STATUS - Status Register . . . 602
21.5.20 EUSART_IF - Interrupt Flag Register . . . 604
21.5.21 EUSART_IEN - Interrupt Enable Register . . . 606
21.5.22 EUSART_SYNCBUSY - Synchronization Busy Register . . . 608
21.5.23 EUSART_DALICFG - DALI Config Register . . . 610

22. I2C - Inter-Integrated Circuit Interface . . . 612
22.1 Introduction . . . 612
22.2 Features . . . 612
22.3 Functional Description . . . 613
22.3.1 Power Domains and Energy Modes . . . 613
22.3.2 I2C-Bus Overview . . . 614
22.3.3 Enable and Reset . . . 618
22.3.4 Pin Configuration . . . 618
22.3.5 Safely Disabling and Changing Follower Configuration . . . 618
22.3.6 Clock Generation . . . 619
22.3.7 Arbitration . . . 619
22.3.8 FIFOs . . . 619
22.3.9 Leader Operation . . . 621
22.3.10 Bus States . . . 629
22.3.11 Follower Operation . . . 629
22.3.12 Transfer Automation . . . 633
22.3.13 Using 10-bit Addresses . . . 634
22.3.14 Error Handling . . . 634
22.3.15 DMA Support . . . 636
22.3.16 Interrupts . . . 636
22.3.17 Wake-up . . . 636
22.4 I2C Register Map . . . 637
22.5 I2C Register Description . . . 638
22.5.1 I2C_IPVERSION - IP VERSION Register . . . 638
22.5.2 I2C_EN - Enable Register . . . 639
22.5.3 I2C_SWRST - SWRST Register . . . 639
22.5.4 I2C_CTRL - Control Register . . . 640
22.5.5 I2C_CMD - Command Register . . . 643
22.5.6 I2C_STATE - State Register . . . 644
22.5.7 I2C_STATUS - Status Register . . . 645
22.5.8 I2C_CLKDIV - Clock Division Register . . . 646
22.5.9 I2C_SADDR - Follower Address Register . . . 646
22.5.10 I2C_SADDRMASK - Follower Address Mask Register . . . 647
22.5.11 I2C_RXDATA - Receive FIFO Data Register . . . 647
22.5.12 I2C_RXDATAP - Receive FIFO Data Peek Register . . . 648
22.5.13 I2C_TXDATA - Transmit FIFO Data Register . . . 648
22.5.14 I2C_IF - Interrupt Flag Register . . . 649
22.5.15 I2C_IEN - Interrupt Enable Register . . . 651
22.5.16 I2C_SYNCBUSY - Syncbusy Status . . . 653
23. ADC - Analog to Digital Converter . . . 655
23.1 Introduction . . . 655
23.2 Features . . . 656
23.3 Functional Description . . . 657
23.3.1 Register Access . . . 657
23.3.2 Clocking . . . 658

23.3.3 Acquisition and Conversion Timing . . . 659
23.3.4 Reference Selection and Analog Gain . . . 663
23.3.5 Input and Configuration Selection . . . 664
23.3.6 Gain and Offset Correction . . . 669
23.3.7 Output Data FIFOs . . . 670
23.3.8 Window Compare . . . 673
23.3.9 Interrupts . . . 674
23.4 ADC Register Map . . . 675
23.5 ADC Register Description . . . 677
23.5.1 ADC_IPVERSION - IPVERSION . . . 677
23.5.2 ADC_EN - Enable . . . 678
23.5.3 ADC_CTRL - Control . . . 679
23.5.4 ADC_CMD - Command . . . 681
23.5.5 ADC_TIMER - Timer Control . . . 682
23.5.6 ADC_STATUS - Status . . . 683
23.5.7 ADC_MASKREQ - Select Channels to Be Converted . . . 684
23.5.8 ADC_STMASK - Scan Table Mask . . . 684
23.5.9 ADC_CMPTHR - Digital Window Comparator Threshold . . . 685
23.5.10 ADC_IF - Interrupt Flags . . . 686
23.5.11 ADC_IEN - Interrupt Enable . . . 687
23.5.12 ADC_TRIGGER - Trigger . . . 688
23.5.13 ADC_CFGx - Configuration . . . 690
23.5.14 ADC_OFFSETSEx - Offset Trimming for SE Mode . . . 691
23.5.15 ADC_OFFSETDIFFx - Offset Trimming for Diff Mode . . . 691
23.5.16 ADC_GAINCALx - Gain Calibration . . . 692
23.5.17 ADC_SCANFIFOCFG - Scan FIFO Configuration . . . 693
23.5.18 ADC_SCANFIFODATA - Scan FIFO Read Data . . . 694
23.5.19 ADC_SCANFIFOSTAT - Scan FIFO Status . . . 695
23.5.20 ADC_SCANDATA - Scan Data . . . 695
23.5.21 ADC_SCANx - SCAN ENTRY . . . 696

24. ACMP - Analog Comparator . . . 697

24.1 Introduction . . . 697
24.2 Features . . . 697
24.3 Functional Description . . . 698
24.3.1 Configuration and Control . . . 699
24.3.2 Warmup Time . . . 699
24.3.3 Response Time . . . 699
24.3.4 Hysteresis . . . 700
24.3.5 VREFDIV Sources . . . 700
24.3.6 Supply Voltage Monitoring (VSENSE) . . . 700
24.3.7 Input Range, Common Mode, and Accuracy Settings . . . 701
24.3.8 Interrupts and PRS Output . . . 701
24.3.9 Output to GPIO . . . 701
24.4 ACMP Register Map . . . 702
24.5 ACMP Register Description . . . 703
24.5.1 ACMP_IPVERSION - IP Version ID . . . 703

24.5.2 ACMP_EN - ACMP Enable . . . 703
24.5.3 ACMP_SWRST - Software Reset . . . 704
24.5.4 ACMP_CFG - Configuration Register . . . 705
24.5.5 ACMP_CTRL - Control Register . . . 706
24.5.6 ACMP_INPUTCTRL - Input Control Register . . . 707
24.5.7 ACMP_STATUS - Status Register . . . 712
24.5.8 ACMP_IF - Interrupt Flag Register . . . 713
24.5.9 ACMP_IEN - Interrupt Enable Register . . . 714
24.5.10 ACMP_SYNCBUSY - Syncbusy . . . 714

25. GPIO - General Purpose Input/Output . . . 715

25.1 Introduction . . . 715
25.2 Features . . . 716
25.3 Functional Description . . . 717
25.3.1 Pin Configuration . . . 718
25.3.2 Low Noise Mode . . . 720
25.3.3 Alternate Port Control . . . 720
25.3.4 Slew Rate . . . 720
25.3.5 Input Disable . . . 720
25.3.6 Configuration Lock . . . 720
25.3.7 EM4 Functionality . . . 720
25.3.8 EM4 Wakeup . . . 721
25.3.9 Debug Connections . . . 721
25.3.10 Interrupt Generation . . . 723
25.3.11 Output to PRS . . . 724
25.3.12 Peripheral Resource Routing . . . 724
25.4 Synchronization . . . 731
25.5 GPIO Register Map . . . 732
25.6 GPIO Register Description . . . 755
25.6.1 GPIO_IPVERSION - Main . . . 755
25.6.2 GPIO_PORTA_CTRL - Port Control . . . 756
25.6.3 GPIO_PORTA_MODEL - Mode Low . . . 757
25.6.4 GPIO_PORTA_MODEH - Mode High . . . 762
25.6.5 GPIO_PORTA_DOUT - Data Out . . . 763
25.6.6 GPIO_PORTA_DIN - Data in . . . 763
25.6.7 GPIO_PORTA_AMUXMODE - AMUXMODE . . . 763
25.6.8 GPIO_PORTB_CTRL - Port Control . . . 764
25.6.9 GPIO_PORTB_MODEL - Mode Low . . . 765
25.6.10 GPIO_PORTB_DOUT - Data Out . . . 767
25.6.11 GPIO_PORTB_DIN - Data in . . . 767
25.6.12 GPIO_PORTB_AMUXMODE - AMUXMODE . . . 768
25.6.13 GPIO_PORTC_CTRL - Port Control . . . 768
25.6.14 GPIO_PORTC_MODEL - Mode Low . . . 769
25.6.15 GPIO_PORTC_DOUT - Data Out . . . 773
25.6.16 GPIO_PORTC_DIN - Data in . . . 773
25.6.17 GPIO_PORTC_AMUXMODE - AMUXMODE . . . 774
25.6.18 GPIO_PORTD_CTRL - Port Control . . . 774

25.6.19 GPIO_PORTD_MODEL - Mode Low . . . 775
25.6.20 GPIO_PORTD_DOUT - Data Out . . . 779
25.6.21 GPIO_PORTD_DIN - Data in . . . 779
25.6.22 GPIO_PORTD_AMUXMODE - AMUXMODE . . . 779
25.6.23 GPIO_LOCK - Lock Register . . . 780
25.6.24 GPIO_GPIOLOCKSTATUS - Lock Status . . . 780
25.6.25 GPIO_ABUSALLOC - A Bus Allocation . . . 781
25.6.26 GPIO_BBUSALLOC - B Bus Allocation . . . 783
25.6.27 GPIO_CDBUSALLOC - CD Bus Allocation . . . 785
25.6.28 GPIO_EXTIPSELL - External Interrupt Port Select Low . . . 787
25.6.29 GPIO_EXTIPSELH - External Interrupt Port Select High . . . 790
25.6.30 GPIO_EXTIPINSELL - External Interrupt Pin Select Low . . . 792
25.6.31 GPIO_EXTIPINSELH - External Interrupt Pin Select High . . . 795
25.6.32 GPIO_EXTIRISE - External Interrupt Rising Edge Trigger . . . 796
25.6.33 GPIO_EXTIFALL - External Interrupt Falling Edge Trigger . . . 796
25.6.34 GPIO_IF - Interrupt Flag . . . 797
25.6.35 GPIO_IEN - Interrupt Enable . . . 799
25.6.36 GPIO_EM4WUEN - EM4 Wakeup Enable . . . 800
25.6.37 GPIO_EM4WUPOL - EM4 Wakeup Polarity . . . 801
25.6.38 GPIO_DBGROUTEPEN - Debugger Route Pin Enable . . . 802
25.6.39 GPIO_TRACEROUTEPEN - Trace Route Pin Enable . . . 803
25.6.40 GPIO_ACMP0_ROUTEEN - ACMP0 Pin Enable . . . 803
25.6.41 GPIO_ACMP0_ACMPOUTROUTE - ACMPOUT Port/Pin Select . . . 804
25.6.42 GPIO_ACMP1_ROUTEEN - ACMP1 Pin Enable . . . 804
25.6.43 GPIO_ACMP1_ACMPOUTROUTE - ACMPOUT Port/Pin Select . . . 805
25.6.44 GPIO_EUSART0_ROUTEEN - EUSART0 Pin Enable . . . 805
25.6.45 GPIO_EUSART0_CSROUTE - CS Port/Pin Select . . . 806
25.6.46 GPIO_EUSART0_CTSROUTE - CTS Port/Pin Select . . . 806
25.6.47 GPIO_EUSART0_RTSROUTE - RTS Port/Pin Select . . . 807
25.6.48 GPIO_EUSART0_RXROUTE - RX Port/Pin Select . . . 807
25.6.49 GPIO_EUSART0_SCLKROUTE - SCLK Port/Pin Select . . . 808
25.6.50 GPIO_EUSART0_TXROUTE - TX Port/Pin Select . . . 808
25.6.51 GPIO_EUSART1_ROUTEEN - EUSART1 Pin Enable . . . 809
25.6.52 GPIO_EUSART1_CSROUTE - CS Port/Pin Select . . . 809
25.6.53 GPIO_EUSART1_CTSROUTE - CTS Port/Pin Select . . . 810
25.6.54 GPIO_EUSART1_RTSROUTE - RTS Port/Pin Select . . . 810
25.6.55 GPIO_EUSART1_RXROUTE - RX Port/Pin Select . . . 811
25.6.56 GPIO_EUSART1_SCLKROUTE - SCLK Port/Pin Select . . . 811
25.6.57 GPIO_EUSART1_TXROUTE - TX Port/Pin Select . . . 812
25.6.58 GPIO_EUSART2_ROUTEEN - EUSART2 Pin Enable . . . 812
25.6.59 GPIO_EUSART2_CSROUTE - CS Port/Pin Select . . . 813
25.6.60 GPIO_EUSART2_CTSROUTE - CTS Port/Pin Select . . . 813
25.6.61 GPIO_EUSART2_RTSROUTE - RTS Port/Pin Select . . . 814
25.6.62 GPIO_EUSART2_RXROUTE - RX Port/Pin Select . . . 814
25.6.63 GPIO_EUSART2_SCLKROUTE - SCLK Port/Pin Select . . . 815
25.6.64 GPIO_EUSART2_TXROUTE - TX Port/Pin Select . . . 815
25.6.65 GPIO_FRC_ROUTEEN - FRC Pin Enable . . . 816
25.6.66 GPIO_FRC_DCLKROUTE - DCLK Port/Pin Select . . . 816

25.6.67 GPIO_FRC_DFRAMEROUTE - DFRAME Port/Pin Select . . . 817
25.6.68 GPIO_FRC_DOUTROUTE - DOUT Port/Pin Select . . . 817
25.6.69 GPIO_CMU_ROUTEEN - HOSTCMU Pin Enable . . . 818
25.6.70 GPIO_CMU_CLKIN0ROUTE - CLKIN0 Port/Pin Select . . . 818
25.6.71 GPIO_CMU_CLKOUT0ROUTE - CLKOUT0 Port/Pin Select . . . 819
25.6.72 GPIO_CMU_CLKOUT1ROUTE - CLKOUT1 Port/Pin Select . . . 819
25.6.73 GPIO_CMU_CLKOUT2ROUTE - CLKOUT2 Port/Pin Select . . . 820
25.6.74 GPIO_I2C0_ROUTEEN - I2C0 Pin Enable . . . 820
25.6.75 GPIO_I2C0_SCLROUTE - SCL Port/Pin Select . . . 821
25.6.76 GPIO_I2C0_SDAROUTE - SDA Port/Pin Select. . . . 821
25.6.77 GPIO_I2C1_ROUTEEN - I2C1 Pin Enable . . . 822
25.6.78 GPIO_I2C1_SCLROUTE - SCL Port/Pin Select . . . 822
25.6.79 GPIO_I2C1_SDAROUTE - SDA Port/Pin Select. . . . 823
25.6.80 GPIO_I2C2_ROUTEEN - I2C2 Pin Enable . . . 823
25.6.81 GPIO_I2C2_SCLROUTE - SCL Port/Pin Select . . . 824
25.6.82 GPIO_I2C2_SDAROUTE - SDA Port/Pin Select. . . . 824
25.6.83 GPIO_LETIMER0_ROUTEEN - LETIMER0 Pin Enable . . . 825
25.6.84 GPIO_LETIMER0_OUT0ROUTE - OUT0 Port/Pin Select . . . 825
25.6.85 GPIO_LETIMER0_OUT1ROUTE - OUT1 Port/Pin Select . . . 826
25.6.86 GPIO_LFXO_ROUTEEN - LFXO Pin Enable . . . 826
25.6.87 GPIO_LFXO_LFXOCLKLVRAWROUTE - LFXOCLKLVRAW Port/Pin Select . . . 827
25.6.88 GPIO_MODEM_ROUTEEN - MODEM Pin Enable . . . 828
25.6.89 GPIO_MODEM_ANT0ROUTE - ANT0 Port/Pin Select . . . 829
25.6.90 GPIO_MODEM_ANT1ROUTE - ANT1 Port/Pin Select . . . 829
25.6.91 GPIO_MODEM_ANTROLLOVERROUTE - ANTROLLOVER Port/Pin Select . . . 830
25.6.92 GPIO_MODEM_ANTRR0ROUTE - ANTRR0 Port/Pin Select . . . 830
25.6.93 GPIO_MODEM_ANTRR1ROUTE - ANTRR1 Port/Pin Select . . . 831
25.6.94 GPIO_MODEM_ANTRR2ROUTE - ANTRR2 Port/Pin Select . . . 831
25.6.95 GPIO_MODEM_ANTRR3ROUTE - ANTRR3 Port/Pin Select . . . 832
25.6.96 GPIO_MODEM_ANTRR4ROUTE - ANTRR4 Port/Pin Select . . . 832
25.6.97 GPIO_MODEM_ANTRR5ROUTE - ANTRR5 Port/Pin Select . . . 833
25.6.98 GPIO_MODEM_ANTSWENROUTE - ANTSWEN Port/Pin Select . . . 833
25.6.99 GPIO_MODEM_ANTSWUSROUTE - ANTSWUS Port/Pin Select . . . 834
25.6.100 GPIO_MODEM_ANTTRIGROUTE - ANTTRIG Port/Pin Select . . . 834
25.6.101 GPIO_MODEM_ANTTRIGSTOPROUTE - ANTTRIGSTOP Port/Pin Select . . . 835
25.6.102 GPIO_MODEM_DCLKROUTE - DCLK Port/Pin Select . . . 835
25.6.103 GPIO_MODEM_DINROUTE - DIN Port/Pin Select . . . 836
25.6.104 GPIO_MODEM_DOUTROUTE - DOUT Port/Pin Select. . . . 836
25.6.105 GPIO_PCNT0_S0INROUTE - S0IN Port/Pin Select . . . 837
25.6.106 GPIO_PCNT0_S1INROUTE - S1IN Port/Pin Select . . . 837
25.6.107 GPIO_PIXELRZ0_ROUTEEN - PIXELRZ0 Pin Enable . . . 838
25.6.108 GPIO_PIXELRZ0_RZTXOUTROUTE - RZTXOUT Port/Pin Select . . . 838
25.6.109 GPIO_PIXELRZ1_ROUTEEN - PIXELRZ1 Pin Enable . . . 839
25.6.110 GPIO_PIXELRZ1_RZTXOUTROUTE - RZTXOUT Port/Pin Select . . . 839
25.6.111 GPIO_PRS0_ROUTEEN - PRS0 Pin Enable . . . 840
25.6.112 GPIO_PRS0_ASYNCH0ROUTE - ASYNCH0 Port/Pin Select . . . 841
25.6.113 GPIO_PRS0_ASYNCH1ROUTE - ASYNCH1 Port/Pin Select . . . 841
25.6.114 GPIO_PRS0_ASYNCH2ROUTE - ASYNCH2 Port/Pin Select . . . 842

25.6.115 GPIO_PRS0_ASYNCH3ROUTE - ASYNCH3 Port/Pin Select . . . 842
25.6.116 GPIO_PRS0_ASYNCH4ROUTE - ASYNCH4 Port/Pin Select . . . 843
25.6.117 GPIO_PRS0_ASYNCH5ROUTE - ASYNCH5 Port/Pin Select . . . 843
25.6.118 GPIO_PRS0_ASYNCH6ROUTE - ASYNCH6 Port/Pin Select . . . 844
25.6.119 GPIO_PRS0_ASYNCH7ROUTE - ASYNCH7 Port/Pin Select . . . 844
25.6.120 GPIO_PRS0_ASYNCH8ROUTE - ASYNCH8 Port/Pin Select . . . 845
25.6.121 GPIO_PRS0_ASYNCH9ROUTE - ASYNCH9 Port/Pin Select . . . 845
25.6.122 GPIO_PRS0_ASYNCH10ROUTE - ASYNCH10 Port/Pin Select . . . 846
25.6.123 GPIO_PRS0_ASYNCH11ROUTE - ASYNCH11 Port/Pin Select . . . 846
25.6.124 GPIO_PRS0_SYNCH0ROUTE - SYNCH0 Port/Pin Select . . . 847
25.6.125 GPIO_PRS0_SYNCH1ROUTE - SYNCH1 Port/Pin Select . . . 847
25.6.126 GPIO_PRS0_SYNCH2ROUTE - SYNCH2 Port/Pin Select . . . 848
25.6.127 GPIO_PRS0_SYNCH3ROUTE - SYNCH3 Port/Pin Select . . . 848
25.6.128 GPIO_RAC_ROUTEEN - RAC Pin Enable . . . 849
25.6.129 GPIO_RAC_LNAENROUTE - LNAEN Port/Pin Select . . . 849
25.6.130 GPIO_RAC_PAENROUTE - PAEN Port/Pin Select . . . 850
25.6.131 GPIO_TIMER0_ROUTEEN - TIMER0 Pin Enable. . . . 850
25.6.132 GPIO_TIMER0_CC0ROUTE - CC0 Port/Pin Select . . . 851
25.6.133 GPIO_TIMER0_CC1ROUTE - CC1 Port/Pin Select . . . 851
25.6.134 GPIO_TIMER0_CC2ROUTE - CC2 Port/Pin Select . . . 852
25.6.135 GPIO_TIMER0_CDTI0ROUTE - CDTI0 Port/Pin Select . . . 852
25.6.136 GPIO_TIMER0_CDTI1ROUTE - CDTI1 Port/Pin Select . . . 853
25.6.137 GPIO_TIMER0_CDTI2ROUTE - CDTI2 Port/Pin Select . . . 853
25.6.138 GPIO_TIMER1_ROUTEEN - TIMER1 Pin Enable. . . . 854
25.6.139 GPIO_TIMER1_CC0ROUTE - CC0 Port/Pin Select . . . 854
25.6.140 GPIO_TIMER1_CC1ROUTE - CC1 Port/Pin Select . . . 855
25.6.141 GPIO_TIMER1_CC2ROUTE - CC2 Port/Pin Select . . . 855
25.6.142 GPIO_TIMER1_CDTI0ROUTE - CDTI0 Port/Pin Select . . . 856
25.6.143 GPIO_TIMER1_CDTI1ROUTE - CDTI1 Port/Pin Select . . . 856
25.6.144 GPIO_TIMER1_CDTI2ROUTE - CDTI2 Port/Pin Select . . . 857
25.6.145 GPIO_TIMER2_ROUTEEN - TIMER2 Pin Enable. . . . 858
25.6.146 GPIO_TIMER2_CC0ROUTE - CC0 Port/Pin Select . . . 859
25.6.147 GPIO_TIMER2_CC1ROUTE - CC1 Port/Pin Select . . . 859
25.6.148 GPIO_TIMER2_CC2ROUTE - CC2 Port/Pin Select . . . 860
25.6.149 GPIO_TIMER2_CC3ROUTE - CC3 Port/Pin Select . . . 860
25.6.150 GPIO_TIMER2_CC4ROUTE - CC4 Port/Pin Select . . . 861
25.6.151 GPIO_TIMER2_CC5ROUTE - CC5 Port/Pin Select . . . 861
25.6.152 GPIO_TIMER2_CC6ROUTE - CC6 Port/Pin Select . . . 862
25.6.153 GPIO_TIMER2_CDTI0ROUTE - CDTI0 Port/Pin Select . . . 862
25.6.154 GPIO_TIMER2_CDTI1ROUTE - CDTI1 Port/Pin Select . . . 863
25.6.155 GPIO_TIMER2_CDTI2ROUTE - CDTI2 Port/Pin Select . . . 863
25.6.156 GPIO_TIMER2_CDTI3ROUTE - CDTI3 Port/Pin Select . . . 864
25.6.157 GPIO_TIMER2_CDTI4ROUTE - CDTI4 Port/Pin Select . . . 864
25.6.158 GPIO_TIMER2_CDTI5ROUTE - CDTI5 Port/Pin Select . . . 865
25.6.159 GPIO_TIMER2_CDTI6ROUTE - CDTI6 Port/Pin Select . . . 865
25.6.160 GPIO_TIMER3_ROUTEEN - TIMER3 Pin Enable. . . . 866
25.6.161 GPIO_TIMER3_CC0ROUTE - CC0 Port/Pin Select . . . 867
25.6.162 GPIO_TIMER3_CC1ROUTE - CC1 Port/Pin Select . . . 867

25.6.163 GPIO_TIMER3_CC2ROUTE - CC2 Port/Pin Select . . . 868
25.6.164 GPIO_TIMER3_CC3ROUTE - CC3 Port/Pin Select . . . 868
25.6.165 GPIO_TIMER3_CC4ROUTE - CC4 Port/Pin Select . . . 869
25.6.166 GPIO_TIMER3_CC5ROUTE - CC5 Port/Pin Select . . . 869
25.6.167 GPIO_TIMER3_CC6ROUTE - CC6 Port/Pin Select . . . 870
25.6.168 GPIO_TIMER3_CDTI0ROUTE - CDTI0 Port/Pin Select . . . 870
25.6.169 GPIO_TIMER3_CDTI1ROUTE - CDTI1 Port/Pin Select . . . 871
25.6.170 GPIO_TIMER3_CDTI2ROUTE - CDTI2 Port/Pin Select . . . 871
25.6.171 GPIO_TIMER3_CDTI3ROUTE - CDTI3 Port/Pin Select . . . 872
25.6.172 GPIO_TIMER3_CDTI4ROUTE - CDTI4 Port/Pin Select . . . 872
25.6.173 GPIO_TIMER3_CDTI5ROUTE - CDTI5 Port/Pin Select . . . 873
25.6.174 GPIO_TIMER3_CDTI6ROUTE - CDTI6 Port/Pin Select . . . 873

26. LDMA - Linked DMA Controller . . . 874

26.1 Introduction . . . 874
26.1.1 Features . . . 875

26.2 Block Diagram . . . 876

26.3 Functional Description . . . 877
26.3.1 Channel Descriptor . . . 877
26.3.2 Channel Configuration . . . 882
26.3.3 Channel Select Configuration . . . 882
26.3.4 Starting a Transfer . . . 882
26.3.5 Managing Transfer Errors . . . 883
26.3.6 Arbitration . . . 883
26.3.7 Channel Descriptor Data Structure . . . 885
26.3.8 Interaction with the EMU . . . 888
26.3.9 Interrupts . . . 888
26.3.10 Debugging . . . 889
26.3.11 Multi-port Transfers . . . 889
26.3.12 Extended Struct . . . 889
26.3.13 Destination Interleaving . . . 890
26.3.14 Disablement . . . 891
26.3.15 Software Reset . . . 891

26.4 Examples . . . 891
26.4.1 Single Direct Register DMA Transfer . . . 892
26.4.2 Descriptor Linked List . . . 893
26.4.3 Single Descriptor Looped Transfer . . . 895
26.4.4 Descriptor List with Looping . . . 896
26.4.5 Simple Inter-Channel Synchronization . . . 897
26.4.6 2D Copy . . . 899
26.4.7 Ping-Pong . . . 901
26.4.8 Scatter-Gather . . . 902

26.5 LDMA Source Selection Details . . . 902
26.5.1 LDMA Source Selection Details . . . 903

26.6 LDMA Register Map . . . 905

26.7 LDMA Register Description . . . 908
26.7.1 LDMA_IPVERSION - IP Version . . . 908

26.7.2 LDMA_EN - Module Enable Disable Register . . . 909
26.7.3 LDMA_SWRST - Software Reset Register . . . 909
26.7.4 LDMA_CTRL - Control Register . . . 910
26.7.5 LDMA_STATUS - Status Register . . . 911
26.7.6 LDMA_SYNCSWSET - Sync Trig Sw Set Register . . . 912
26.7.7 LDMA_SYNCSWCLR - Sync Trig Sw Clear Register . . . 912
26.7.8 LDMA_SYNCHWEN - Sync HW Trigger Enable Register . . . 913
26.7.9 LDMA_SYNCHWSEL - Sync HW Trigger Selection Register . . . 914
26.7.10 LDMA_SYNCSTATUS - Sync Trigger Status Register . . . 914
26.7.11 LDMA_CHEN - Channel Enable Register . . . 915
26.7.12 LDMA_CHDIS - Channel Disable Register . . . 915
26.7.13 LDMA_CHSTATUS - Channel Status Register . . . 916
26.7.14 LDMA_CHBUSY - Channel Busy Register . . . 916
26.7.15 LDMA_CHDONE - Channel Linking Done Register . . . 917
26.7.16 LDMA_DBGHALT - Channel Debug Halt Register . . . 918
26.7.17 LDMA_SWREQ - Channel Software Transfer Request . . . 918
26.7.18 LDMA_REQDIS - Channel Request Disable Register . . . 919
26.7.19 LDMA_REQPEND - Channel Requests Pending Register . . . 919
26.7.20 LDMA_LINKLOAD - Channel Link Load Register . . . 920
26.7.21 LDMA_REQCLEAR - Channel Request Clear Register . . . 920
26.7.22 LDMA_IF - Interrupt Flag Register . . . 921
26.7.23 LDMA_IEN - Done Interrupt Enable Register . . . 922
26.7.24 LDMA_REQABORT - DMA Req Abort . . . 923
26.7.25 LDMA_ABORTSTATUS - Abort Clean Status Register . . . 923
26.7.26 LDMA_CHx_CFG - Configuration . . . 924
26.7.27 LDMA_CHx_LOOP - Loop Counter . . . 925
26.7.28 LDMA_CHx_CTRL - Control . . . 926
26.7.29 LDMA_CHx_SRC - Source Address . . . 928
26.7.30 LDMA_CHx_DST - Destination Address . . . 929
26.7.31 LDMA_CHx_LINK - Link Address . . . 929
26.7.32 LDMA_CHx_XCTRL - Extended Control . . . 930
26.7.33 LDMA_CHx_ILSRC - Interleaving Source Address . . . 930
26.8 LDMAXBAR Register Map . . . 931
26.9 LDMAXBAR Register Description . . . 931
26.9.1 LDMAXBAR_IPVERSION - IP Veersion ID . . . 931
26.9.2 LDMAXBAR_CHx_REQSEL - Channel Peripheral Request Select Reg... . . . 932
27. LED Pre-Driver (LEDDRV) . . . 933
27.1 Introduction . . . 933
27.2 Features . . . 933
27.3 Functional Description . . . 934
27.3.1 Normal BCM Operation . . . 935
27.3.2 ABUS Signal Routing . . . 939
27.3.3 Charge Pump Operation . . . 940
27.3.4 DC Supply Operation . . . 940
27.3.5 Dimming and Mixing . . . 940
27.3.6 Dual Driver Mode . . . 940

27.3.7 Direct Drive Mode . . . 940
27.4 LEDDRV Register Map . . . 941
27.5 LEDDRV Register Description . . . 942
27.5.1 LEDDRV_IPVERSION - IP VERSION Register . . . 942
27.5.2 LEDDRV_EN - Enable Register . . . 942
27.5.3 LEDDRV_SWRST - SWRST Register . . . 943
27.5.4 LEDDRV_CFG - Configuration Register . . . 944
27.5.5 LEDDRV_CTRL - Control Register . . . 945
27.5.6 LEDDRV_STATUS - Status Register . . . 946
27.5.7 LEDDRV_ABUSCTRL0 - LEDDRV ABUS Control 0 . . . 947
27.5.8 LEDDRV_ABUSCTRL1 - LEDDRV ABUS Control 1 . . . 948
27.5.9 LEDDRV_IF - Interrupt Flag Register . . . 949
27.5.10 LEDDRV_IEN - Interrupt Enable Register . . . 950
28. PIXELRZ - Serial Pixel Interface . . . 951
28.1 Introduction . . . 951
28.2 Features . . . 951
28.3 Functional Description . . . 952
28.3.1 Sequential Mode Operation . . . 953
28.3.2 Parallel Mode Operation . . . 954
28.3.3 Clocking . . . 955
28.3.4 Symbol Generation . . . 955
28.3.5 Frame Triggering . . . 956
28.3.6 Frame Formatting . . . 956
28.3.7 FIFO Interface . . . 956
28.3.8 Data Alignment and Bit Order . . . 957
28.3.9 Setup . . . 959
28.4 PIXELRZ Register Map . . . 960
28.5 PIXELRZ Register Description . . . 961
28.5.1 PIXELRZ_IPVERSION - IP VERSION Register . . . 961
28.5.2 PIXELRZ_EN - Enable Register . . . 962
28.5.3 PIXELRZ_SWRST - Software Reset Register . . . 962
28.5.4 PIXELRZ_CFG - Configuration Register . . . 963
28.5.5 PIXELRZ_FRAMECFG - Frame Configuration Register . . . 965
28.5.6 PIXELRZ_T0CFG - Zero Symbol Configuration Register . . . 965
28.5.7 PIXELRZ_T1CFG - One Symbol Configuration Register . . . 966
28.5.8 PIXELRZ_TSTOPCFG - Stop Sympbol Configuration Register . . . 966
28.5.9 PIXELRZ_TRSTCFG - Reset Symbol Configuration Register . . . 967
28.5.10 PIXELRZ_CMD - Command Register . . . 967
28.5.11 PIXELRZ_TXDATA - Transmit Buffer Data Register . . . 968
28.5.12 PIXELRZ_STATUS - Status Register . . . 969
28.5.13 PIXELRZ_IF - Interrupt Flag Register . . . 970
28.5.14 PIXELRZ_IEN - Interrupt Enable Register . . . 970
29. PCNT - Pulse Counter . . . 971
29.1 Introduction . . . 971
29.2 Features . . . 971

29.3 Functional Description . . . 972
29.3.1 Pulse Counter Modes . . . 972
29.3.2 Hysteresis . . . 979
29.3.3 Auxiliary Counter . . . 980
29.3.4 Register Access . . . 980
29.3.5 Clock Sources . . . 980
29.3.6 Input Filter . . . 981
29.3.7 Edge Polarity . . . 981
29.3.8 PRS and PCNTn.S0IN,PCNTn.S1IN Inputs . . . 981
29.3.9 Interrupts . . . 981
29.4 PCNT Register Map . . . 983
29.5 PCNT Register Description . . . 984
29.5.1 PCNT_IPVERSION - IP Version ID . . . 984
29.5.2 PCNT_EN - Module Enable Register . . . 985
29.5.3 PCNT_SWRST - Software Reset Register . . . 985
29.5.4 PCNT_CFG - Configuration Register . . . 986
29.5.5 PCNT_CTRL - Control Register . . . 987
29.5.6 PCNT_CMD - Command Register . . . 988
29.5.7 PCNT_STATUS - Status Register . . . 989
29.5.8 PCNT_IF - Interrupt Flag Register . . . 990
29.5.9 PCNT_IEN - Interrupt Enable Register . . . 991
29.5.10 PCNT_CNT - Counter Value Register . . . 991
29.5.11 PCNT_AUXCNT - Auxiliary Counter Value Register . . . 992
29.5.12 PCNT_TOP - Top Value Register . . . 992
29.5.13 PCNT_TOPB - Counter Top Value Buffer Register . . . 993
29.5.14 PCNT_OVSCTRL - Oversampling Control Register . . . 993
29.5.15 PCNT_SYNCBUSY - Synchronization Busy Register . . . 994
29.5.16 PCNT_LOCK - Configuration Lock Register . . . 994
30. WDOG - Watch Dog Timer . . . 995
30.1 Introduction . . . 995
30.2 Features . . . 995
30.3 Functional Description . . . 995
30.3.1 Power Domains and Energy Modes . . . 996
30.3.2 Clock Source . . . 996
30.3.3 Debug Functionality . . . 996
30.3.4 Energy Mode Handling . . . 996
30.3.5 Warning Interrupt . . . 996
30.3.6 Window Interrupt . . . 997
30.3.7 PRS as Watchdog Clear . . . 998
30.3.8 PRS Rising Edge Monitoring . . . 998
30.4 WDOG Register Map . . . 999
30.5 WDOG Register Description . . . 1000
30.5.1 WDOG_IPVERSION - IP Version Register . . . 1000
30.5.2 WDOG_EN - Enable Register . . . 1000
30.5.3 WDOG_CFG - Configuration Register . . . 1001
30.5.4 WDOG_CMD - Command Register . . . 1003

30.5.5 WDOG_STATUS - Status Register 1004
30.5.6 WDOG_IF - Interrupt Flag Register 1004
30.5.7 WDOG_IEN - Interrupt Enable Register 1005
30.5.8 WDOG_LOCK - Lock Register 1005
30.5.9 WDOG_SYNCBUSY - Synchronization Busy Register 1006
31. Revision History 1007
Appendix A Abbreviations 1009

1. About This Document

1.1 Introduction

This document contains reference material for the SiXG301 devices. All modules and peripherals in the SiXG301 devices are described in general terms. Not all modules are present in all devices and the feature set for each device might vary. Such differences, including specific device pinouts, are covered in the device data sheets.

1.2 Conventions

Register Names

Register names are given with a module name prefix followed by the short register name:

TIMERn_CTRL - Control Register

The "n" denotes the module number for modules which can exist in more than one instance.

Some registers are grouped which leads to a group name following the module prefix:

GPIO_PORTA_DOUT - Port Data Out Register

TIMER_CCx_CTRL

The "x" denotes the different channels.

Bit Fields

Registers contain one or more bit fields which can be 1 to 32 bits wide. Bit fields wider than 1 bit are given with start (x) and stop (y) bit [y:x].

Bit fields containing more than one bit are unsigned integers unless otherwise is specified.

Unspecified bit field settings must not be used, as this may lead to unpredictable behaviour.

Address

The addresses for each register can be found by adding the base address of the module found in the Memory Map (see 4. System Bus and Memory Map), and the offset address for the register (found in module Register Map).

Access Type

The register access types used in the register descriptions are explained in Table 1.1. Register Access Types.

Table 1.1. Register Access Types

Access Type	Description
R	Read only. Writes are ignored
RW	Readable and writable
RW1	Readable and writable. Only writes to 1 have effect
W1	Read value undefined. Only writes to 1 have effect
W	Write only. Read value undefined.
RWH	Readable, writable, and updated by hardware
RW(nB), RWH(nB), etc.	"(nB)" suffix indicates that a bitfield explicitly does not support peripheral bit set/clear/toggle operations (see 4. System Bus and Memory Map)
RW(r), R(r), etc.	"(r)" suffix indicates that reading the register causes an action and may alter the register value.

Number format

0x prefix is used for hexadecimal numbers

0b prefix is used for binary numbers

Numbers without prefix are in decimal representation.

Reserved

Registers and bit fields marked with **reserved** are reserved for future use. These should be written to their reset value unless otherwise stated in the Register Description. Read values for reserved bits may be different in future or prior devices.

Reset Value

The reset value denotes the value after reset.

Registers denoted with X have unknown value out of reset and need to be initialized before use. Note that read-modify-write operations on these registers before they are initialized results in undefined register values.

Pin Connections

Pin connections are given with a module prefix followed by a short pin name:

CMU.CLKOUT1 (Clock management unit, clock output pin number 1)

1.3 Related Documentation

Further documentation on the SiXG301 devices and the ARM Cortex-M33 can be found at the Silicon Labs and ARM web pages:

www.silabs.com

www.arm.com

2. System Overview

Quick Facts

What?

The SiXG301 is a highly integrated, configurable, and low power wireless System-on-Chip (SoC) with a robust set of MCU and radio peripherals and security features.

Why?

The radio enables support for Matter, OpenThread, Zigbee, Bluetooth Low Energy, and proprietary protocols in 2.4 GHz frequency bands while the MCU system allows customized protocols and applications to run efficiently. Secure Vault technology provides remote and local protection against a variety of attacks.

How?

Dynamic or fixed packet lengths, optional address recognition, and flexible CRC and security schemes make the SiXG301 ideal for many wireless IoT applications. High-performance analog and digital peripherals allow complete applications to run on the SiXG301 SoC.

2.1 Introduction

The high level features of SiXG301 include:

- High-performance radio transceiver
  - Low-power consumption in transmit, receive, and standby modes
  - Excellent receiver performance, including sensitivity, selectivity, and blocking
  - Excellent transmitter performance, including programmable output power, low phase noise, and power-amplifier (PA) ramping
- Configurable protocol support, including standards and customer-developed protocols
  - Preamble and frame synchronization insertion in transmit and recovery in receive
  - Flexible CRC support, including configurable polynomial and multiple CRCs for single data frames
  - Basic address filtering performed in hardware
- High-performance, low-power MCU system
  - High-performance 32-bit ARM Cortex-M33 CPU
  - Complete set of digital peripherals
  - Peripheral Reflex System (PRS)
  - Precision analog peripherals
- Low external component count
  - Fully integrated 2.4 GHz BALUN
  - Integrated tunable crystal loading capacitors

A further introduction to the MCU and radio system is included in the following sections.

Note: Detailed performance numbers, current consumption, pinout, etc., are available in the device data sheets.

2.2 Block Diagrams

The block diagram for the SiXG301 System-On-Chip series is shown in (Figure 2.1. SiXG301 System-On-Chip Block Diagram).

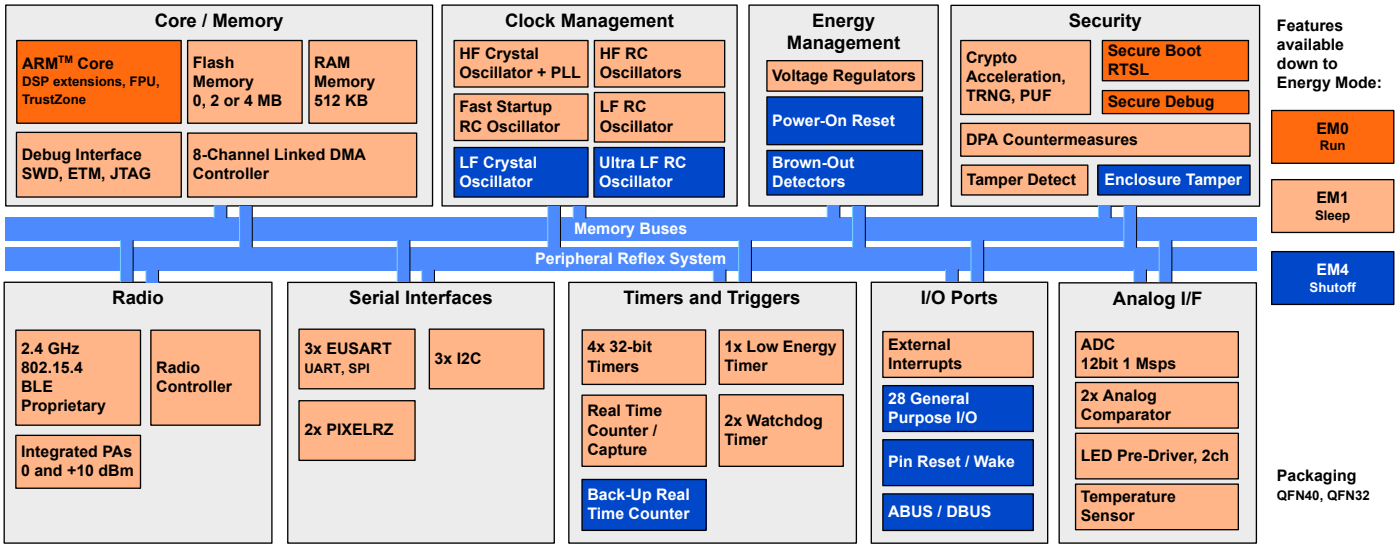


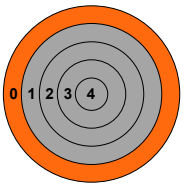
Figure 2.1. SiXG301 System-On-Chip Block Diagram

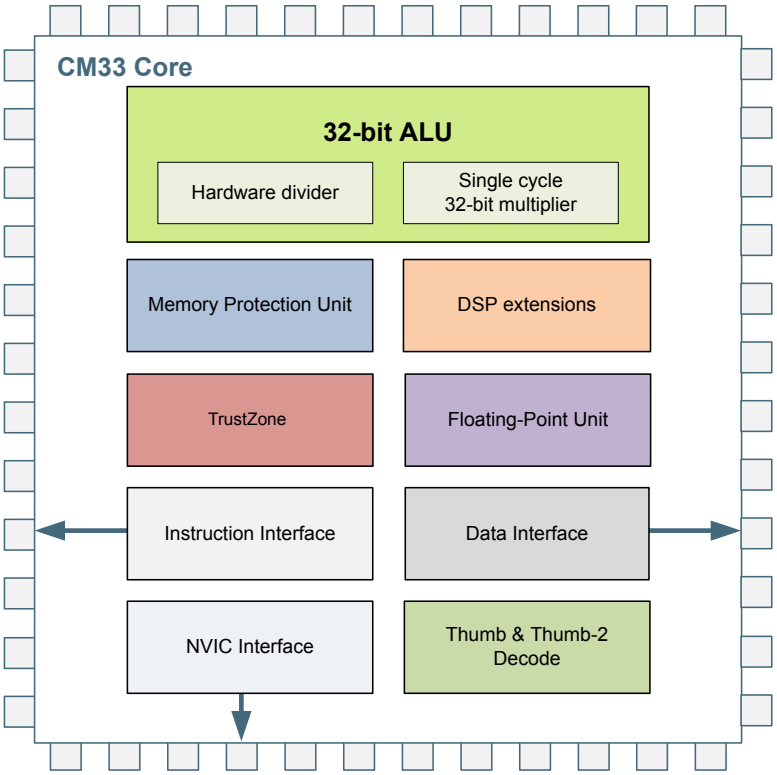
2.3 MCU Features overview

- **ARM Cortex-M33 CPU platform**
  - High-performance 32-bit processor @ up to 150 MHz
  - DSP instruction support and floating-point unit
  - Memory Protection Unit
  - Wake-up Interrupt Controller
- **Flexible Energy Management System**
  - Three Energy Modes, EM0, EM1, and EM4, provide flexibility between higher performance and low power
  - Core system clock can be throttled to lower frequencies when CPU performance is not required, saving energy
  - Brown Out Detection
  - State Retention
- **Up to 4 MB co-packaged flash memory**
- **Up to 512 KB RAM**
- **Up to 28 General Purpose I/O pins**
  - Configurable push-pull, open-drain, pull-up/down, input filter, slew rate
  - Configurable peripheral I/O locations
  - 16 asynchronous external interrupts
  - Output state retention and wake-up from EM4 Shutoff Mode
- **8 Channel DMA Controller**
  - Alternate/primary descriptors with scatter-gather/ping-pong operation
- **16 Channel Peripheral Reflex System (PRS)**
  - Autonomous inter-peripheral signaling enables smart operation in low energy modes
  - 12 asynchronous channels with configurable logic functionality
  - 4 synchronous channels for high-speed signalling between TIMER and IADC
- **Cryptographic Acceleration**
  - AES encryption / decryption, with 128 or 256 bit keys
  - Multiple AES modes of operation, including Counter (CTR), Electronic CodeBook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), Counter mode with CBC-MAC (CCM/CCM*), Cipher-based Message Authentication Code (CMAC), and Galois Message Authenitcation Mode (GCM/GMAC)
  - Accelerated SHA-1 and SHA-2 (SHA-224 / SHA-256)
  - Accelerated Elliptic Curve Cryptography (ECC), with binary or prime fields
  - Flexible 256-bit ALU and sequencer
  - True random number generation
- **General Purpose Cyclic Redundancy Check (GPCRC)**
  - Programmable 16-bit polynomial, fixed 32-bit polynomial
  - The GPCRC module is in addition to the radio CRC
- **Communication interfaces**
  - 3 × Enhanced Universal Synchronous/Asynchronous Receiver/Transmitter (EUSART)
    - UART/SPI/IrDA support
    - High-speed operation
    - Buffered full/half-duplex operation
    - Hardware flow control
    - 7/8/9 data bits
  - 3 × I^2C Interface (I2C) with SMBus support
    - Address recognition
  - 2 × Serial Pixel Interface (PIXELRZ)
    - Autonomously control up to 512 serial-connected individually-addressable LEDs

- **Timers/Counters**
  - 2 × 7-channel and 2 × 3-channel 32-bit Timer/Counters (TIMER)
    - Up to 7 Compare/Capture/PWM channels
    - Three complementary outputs with programmable dead time
    - PWM output dithering
  - 24-bit Low Energy Timer (LETIMER)
  - 16-bit Pulse Counter (PCNT)
  - 32-bit Real-Time Capture Counter (SYSRTC)
  - 32-bit Ultra Low Energy Backup Real Time Counter (BURTC) for periodic wake-up from any Energy Mode
  - 2 × Watchdog Timer (WDOG)
- **Precision analog peripherals**
  - ADC 12-bit resolution at 1 Msps
    - Single-ended or differential operation
    - Hardware accumulator
    - Multi-channel scan controller
  - 2 ×Analog Comparator (ACMP)
    - Programmable speed/current
  - Analog Bus (ABUS) signal routing
  - Accurate die temperature sensor
  - LED Pre-driver (LEDDRV)
    - Directly drive up to two FET gates for LED lighting applications
    - Charge pump generates 6 V from 3.3 V IOVDD power rail
    - AC, current, and drain sense feedback for efficient control
- **Power-on Reset (POR) and Brown-Out Detector (BOD)**
- **Debug Interfaces**
  - 2-pin serial-wire debug (SWD) interface
  - 4-pin Joint Test Action Group (JTAG) interface
  - Embedded Trace Macrocell (ETM)

3. System Processor





Quick Facts

What?

The SiXG301 features the industry leading Cortex-M33 CPU from ARM.

Why?

The ARM Cortex-M33 is designed for exceptionally short response time, high code density, and high 32-bit throughput while maintaining a strict cost and power consumption budget.

How?

Combined with the ultra-low energy peripherals available in SiXG301 devices, the Cortex-M33 processor's Harvard architecture, 3 stage pipeline, single cycle instructions, Thumb-2 instruction set support, and fast interrupt handling make it perfect for 8-bit, 16-bit, and 32-bit applications.

3.1 Introduction

The ARM Cortex-M33 32-bit RISC processor provides outstanding computational performance and exceptional system response to interrupts while meeting low cost requirements and low power consumption.

The ARM Cortex-M33 implemented is revision r0p4.

3.2 Features

- Harvard architecture
  - Separate data and program memory buses (No memory bottleneck as in a single bus system)
- 3-stage pipeline
- Thumb-2 instruction set
  - Enhanced levels of performance, energy efficiency, and code density
- Single cycle multiply and hardware divide instructions
  - 32-bit multiplication in a single cycle
  - Signed and unsigned divide operations between 2 and 11 cycles
- 1.5 DMIPS/MHz
- TrustZone
  - Independent Secure and Privileged states
  - Accelerated context switching
- 16 Region MPU
- 24-bit System Tick Timer for Real Time OS
- Excellent 32-bit migration choice for 8/16 bit architecture based designs
  - Simplified stack-based programmer's model is compatible with traditional ARM architecture and retains the programming simplicity of legacy 8-bit and 16-bit architectures
- Aligned or unaligned data storage and access
  - Contiguous storage of data requiring different byte lengths
  - Data access in a single core access cycle
- Integrated power modes
  - Sleep Now mode for immediate transfer to low power state
  - Sleep on Exit mode for entry into low power state after the servicing of an interrupt
  - Ability to extend power savings to other system components
- Optimized for low latency, nested interrupts

3.3 Functional Description

For a full functional description of the ARM Cortex-M33 implementation in the SiXG301 family, the reader is referred to the ARM Cortex-M33 documentation.

3.3.1 Interrupt Operation

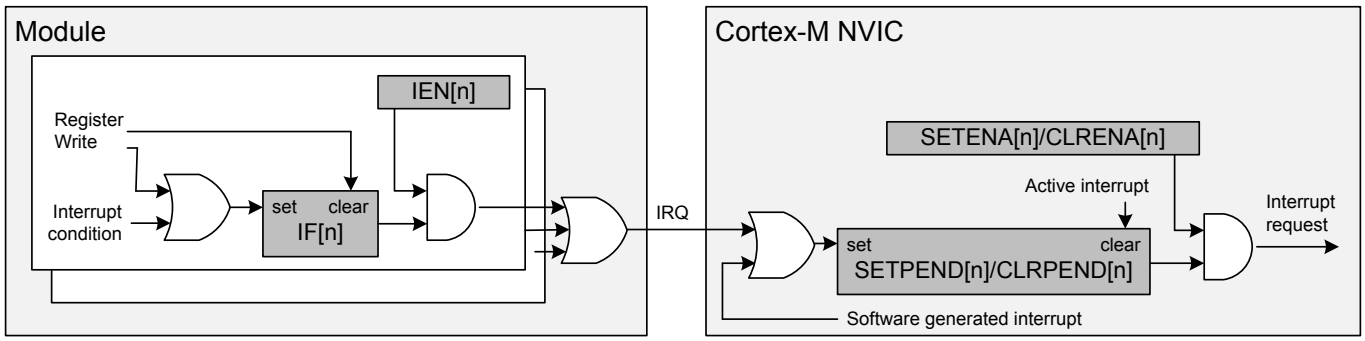


Figure 3.1. Interrupt Operation

The interrupt request (IRQ) lines are connected to the Cortex-M33. Each of these lines (shown in 3.3.3 Interrupt Request Lines (IRQ)) is connected to one or more interrupt flags in one or more modules. The interrupt flags are set by hardware on an interrupt condition. It is also possible to set/clear the interrupt flags through the IF register interface. When setting or clearing an interrupt through the IF register use of the IF_SET or IF_CLR bit operation registers is required; directly writing the main interrupt flag register will have no effect.

Each interrupt flag is then qualified with its own interrupt enable bit (IEN register), before being OR'ed with the other interrupt flags to generate the IRQ. A high IRQ line will set the corresponding pending bit (can also be set/cleared with the SETPEND/CLRPEND bits in ISPRn/ICPRn) in the Cortex-M33 NVIC. The pending bit is then qualified with an enable bit (set/cleared with SETENA/CLRENA bits in ISERn/ICERn) before generating an interrupt request to the core. Figure 3.1. Interrupt Operation illustrates the interrupt system. For more information on how the interrupts are handled inside the Cortex-M33, the reader is referred to the **ARM Cortex-M33 Processor Technical Reference Manual**.

3.3.1.1 Avoiding Extraneous Interrupts

There can be latencies in the system such that clearing an interrupt flag could take longer than leaving an Interrupt Service Routine (ISR). This can lead to the ISR being re-entered as the interrupt flag has yet to clear immediately after leaving the ISR. To avoid this, when clearing an interrupt flag at the end of an ISR, the user should execute ARM's Data Synchronization Barrier (DSB) instruction. Another approach is to clear the interrupt flag immediately after identifying the interrupt source and then service the interrupt as shown in the pseudo-code below. The ISR typically is sufficiently long to more than cover the few cycles it may take to clear the interrupt status, and also allows the status to be checked for further interrupts before exiting the ISR.

```
irqXServiceRoutine() {
  do {
    clearIrqXStatus();
    serviceIrqX();
  } while(irqXStatusIsActive());
}
```

3.3.2 TrustZone

The Cortex-M33 implements ARM TrustZone which provides the ability to restrict access to peripherals and memory regions based on the CPU security attribute. TrustZone works in combination which the MPU which controls privileged/unprivileged execution of code to provide a full security solution. The Security Management Unit (SMU) is used to configure access restrictions in the various modes. Refer to 10. SMU - Security Management Unit for more information.

For information about TrustZone features in the core or information on TrustZone specific instructions please see the **ARM Cortex-M33 Processor Technical Reference Manual** provided by ARM

3.3.3 Interrupt Request Lines (IRQ)

This table shows all IRQ's for the system processor. All interrupts are M33 High Speed interrupts.

See the individual peripheral chapters for more information on interrupt function.

IRQ #	Name
0	SETAMPERHOST
1	SEMBRX
2	SEMBTX
3	SMU_SECURE
4	SMU_S_PRIVILEGED
5	SMU_NS_PRIVILEGED
6	EMU
7	EMUDG
8	SYSMBLPW0CPU
9	ETAMPDET
10	TIMER0
11	TIMER1
12	TIMER2
13	TIMER3
14	SYSRTC_SEQ
15	SYSRTC_APP
16	SYSRTC_MS
17	EUSART0_RX
18	EUSART0_TX
19	EUSART1_RX
20	EUSART1_TX
21	EUSART2_RX
22	EUSART2_TX
23	L1ICACHE0
24	L2ICACHE0
25	BURTC
26	LETIMER0
27	PIXELRZ0
28	PIXELRZ1
29	SYSCFG
30	DMEM
31	LDMA0_CHNL0
32	LDMA0_CHNL1
33	LDMA0_CHNL2
34	LDMA0_CHNL3
35	LDMA0_CHNL4
36	LDMA0_CHNL5
37	LDMA0_CHNL6

IRQ #	Name
38	LDMA0_CHNL7
39	LFXO
40	LFRCO
41	ULFRCO
42	GPIO_ODD
43	GPIO_EVEN
44	I2C0
45	I2C1
46	I2C2
47	BUFC
48	FRC_PRI
49	FRC
50	PROTIMER
51	RAC_RSM
52	RAC_SEQ
53	SYNTH
54	RFECA0
55	RFECA1
56	MODEM
57	AGC
58	RFTIMER
59	SEQACC
60	HFRCOLPW
61	HFRCODPLLLPW
62	ACMP0
63	ACMP1
64	WDOG0
65	WDOG1
66	HFXO0
67	HFRCO0
68	HFRCOEM23
69	CMU
70	RPA
71	KSURPA
72	KSULPWAES
73	KSUHOSTSYMCRYPTO
74	SYMCRYPTO
75	LPWAES
76	ADC0
77	LEDDRV0
78	DPLL0

IRQ #	Name
79	SOCPLL0
80	PCNT0
81	SW0
82	SW1
83	SW2
84	SW3
85	KERNEL0
86	KERNEL1
87	M33CTI0
88	M33CTI1
89	FPUEXH

4. System Bus and Memory Map

4.1 Introduction

The SiXG301 includes on-chip RAM and external flash memory interface hardware. It uses AMBA buses to move data between peripherals, memory, and the processor cores. All memories and register interfaces are memory mapped into a unified system address space.

4.2 Functional Description

The internal memory segments of the Cortex-M33 are mapped into the system memory map as shown by Table 4.1. Host Core Address Map.

Table 4.1. Host Core Address Map

Non-Secure Address Range	Secure Address Range	Region
0x00800000 - 0x0087FFFFF	0x10800000 - 0x1087FFFFF	DMEM (CODE Bus)[1, 2]
0x01000000 - 0x04FFFFFF	0x11000000 - 0x14FFFFFF	EXTMEM (CODE Bus)[1, 3]
0x0FE00000 - 0x0FE0BFFF	0x1FE00000 - 0x1FE0BFFF[4]	DEVINFO[4]
0x20000000 - 0x2007FFFF	0x30000000 - 0x3007FFFF	DMEM (SYSTEM Bus)[2]
0x40000000 - 0x403FFFFF	0x50000000 - 0x503FFFFF	APBLF Peripherals
0x40800000 - 0x40BFFFFF	0x50800000 - 0x50BFFFFF	APBHF Peripherals
0x41000000 - 0x413FFFFF	0x51000000 - 0x513FFFFF	APBUHF Peripherals
0x41C00000 - 0x41FFFFFF	0x51C00000 - 0x51FFFFFF	KSU
0x42800000 - 0x42BFFFFF	0x52800000 - 0x52BFFFFF	SEMAILBOX
0x61000000 - 0x67FFFFFF	0x71000000 - 0x77FFFFFF	EXTMEM (SYSTEM Bus)[3]
0xA0000000 - 0xA03FFFFF	0xB0000000 - 0xB03FFFFF	LPW0 Radio
0xA3000000 - 0xA307FFFF	0xB3000000 - 0xB307FFFF	APBSYSMB Peripherals

Note:

1. DMEM and EXTMEM have separate aliased regions on the CODE and SYSTEM buses. Addresses 0x00000000 to 0x1FFFFFFF are accessed using the CODE bus, and addresses 0x20000000 and above are accessed using the SYSTEM bus. All regions are cachable by default, but can be declared non-cacheable using the MPU.
2. DMEM CODE regions allow for optimized instruction execution from DMEM, while the host data interface targets the DMEM SYSTEM region for stack and variable storage.
3. Flash memory partitions designated as code space should target the CODE region. Flash memory partitions designated as non-volatile data storage may target the EXTMEM SYSTEM data region.
4. The DEVINFO region is always non-secure access in both aliased locations.

Most areas are aliased into secure access and non-secure access address ranges, while DMEM and EXTMEM also have aliased regions mapped into CODE and SYSTEM space.

EXTMEM maps the external code memory into addressable space. The CODE aliases are intended for program code execution, while SYSTEM aliases may be used to access stored data. The allocation of EXTMEM for different levels of encryption and authentication is managed by the SE interface.

DMEM maps the system RAM into addressable space. As with EXTMEM, CODE aliases are intended for program code stored in RAM, while SYSTEM aliases are primarily intended for data storage and program stack.

DEVINFO space is an area which provides quick access to the DEVINFO BASE fragment. Most of the information available in DEVINFO is accessed by requesting the containing fragment via the Secure Engine. The DEVINFO area in the memory map is non-secure access at all aliases.

Peripheral interfaces are mapped to the APBLF, APBHF, APBUHF, and APBSYSMB spaces. See 4.2.4 Peripherals for additional information. The other mapped regions include the SEMAILBOX, LPW0 Radio, and KSU.

4.2.1 Bus Matrix

A multilayer AMBA AXI bus matrix connects the manager bus interfaces to subordinates. The bus matrix allows several subordinates to be accessed simultaneously. An AMBA APB interface is used for the peripherals, which are accessed through a bridge connected to the AXI bus matrix.

The Host Core has two bus managers (Code and System) so that it may retrieve instructions and data in parallel.

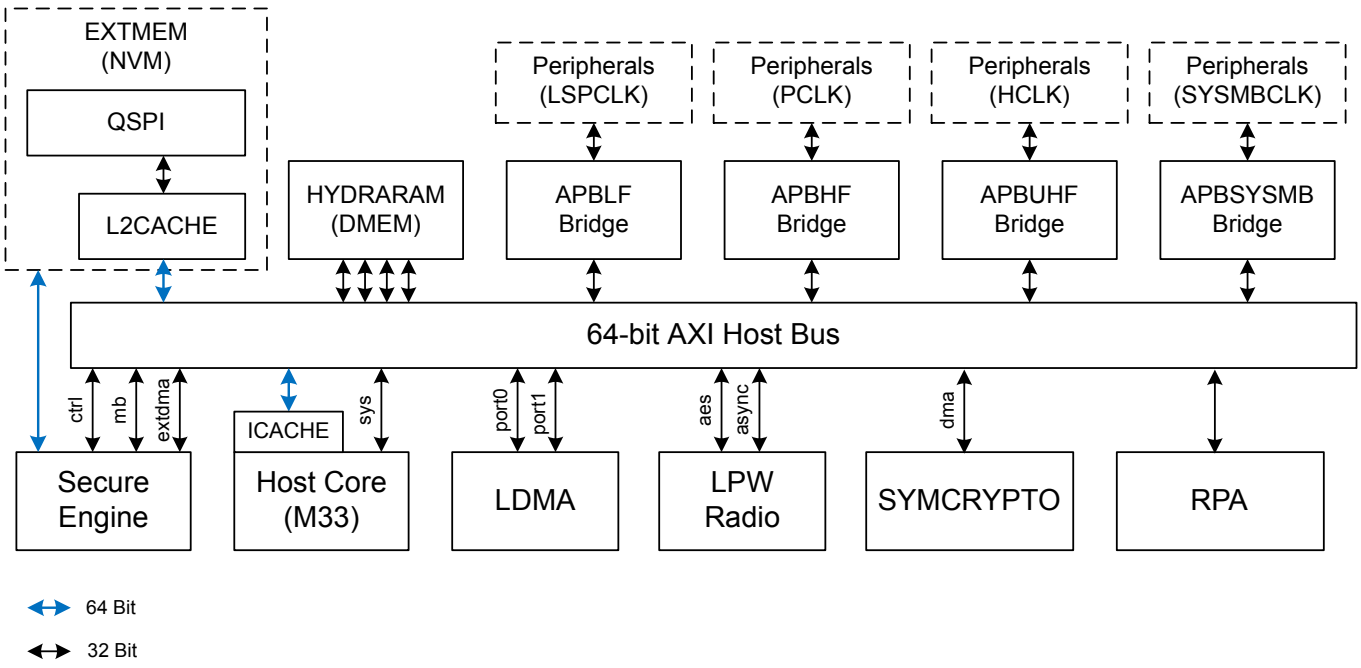


Figure 4.1. SiXG301 Host Bus System

4.2.1.1 Arbitration

The Bus Matrix uses a round-robin arbitration algorithm which enables high throughput and low latency, while starvation of simultaneous accesses to the same bus subordinate are eliminated. Round-robin does not assign a fixed priority to each bus manager.

4.2.1.2 Bus Faults

System accesses from the core can receive a bus fault in the following condition(s):

- The core attempts to access an address that is not assigned to any peripheral or other system device. These faults can be enabled or disabled by setting the ADDRFAULTEN bit in the SYSCFG_CTRL register.
- The HYDRARAM controller detects an access to an address that is not allocated in DMEM space. These faults can be enabled or disabled by setting the ADDRFAULTEN bit in the HYDRARAM_CTRL register.
- The core attempts to access a peripheral or system device that has its clock disabled. This fault can be enabled or disabled by setting the CLKDISFAULTEN bit in the SYSCFG_CTRL register.
- The core attempts to access a peripheral register that is LOCKED.
- The HYDRARAM controller detects a 2-bit ECC error. These faults can be enabled or disabled by setting the ECCERRFAULTEN bit in the HYDRARAM_CTRL register
- One of the SMU blocks (PPU, BMPU, MSPU) encounter a security-level access fault. These blocks have individual control bits to generate a bus fault or respond with RAZ/WI.
- Registers with synchronization requirements may generate bus faults if accessed incorrectly. See 4.2.4.4 Peripheral Access Performance for more details on register access types. In particular the following actions can cause bus faults:
  - CONFIG register written while peripheral enabled.
  - SYNC register written while peripheral disabled
  - LFSYNC register written while a previous write is pending
  - Peripheral disabled while any LFSYNC write is pending

In addition to any condition-specific bus fault control bits, the bus fault exception itself can be enabled or disabled in the same way as all other internal core exceptions.

4.2.2 External Memory

External flash memory is accessed through the EXTMEM subsystem. EXTMEM controls memory access through a high speed quad SPI interface (QSPI) with dedicated encryption, decryption and authentication hardware. The L2CACHE caches read data from the external memory to improve system efficiency.

Flash memory contents are managed by the SE and the EXTMEM hardware to provide robust security for program memory. The user can partition the flash memory into code and non-volatile data storage segments according to the needs of the application.

- Up to 4 MB of co-packaged memory
- Support for full encryption and authentication
- Encrypted eXecute-in-Place (EXiP) of encrypted application code
- Authenticated eXecute-in-Place (AXiP) for authenticated and encrypted application code
- Storage of non-volatile data

4.2.3 RAM

The primary task of the RAM memory is to store application data. Additionally, it is possible to execute instructions from RAM, and the DMA may be set up to transfer data between the RAM, EXTMEM and peripherals.

The device contains several blocks of SRAM for various purposes including general DMEM data memory and various LPW radio subsystem RAMs. A multi-port RAM controller (HYDRARAM) manages accesses to the DMEM space.

- Up to 512 KB of RAM in DMEM
- RAM blocks may be powered down when not in use
- Support for sequential and interleaved addressing

4.2.4 Peripherals

The peripherals are mapped into the peripheral memory segments, each with a fixed size address range shown in the 4.2.4.1 Peripheral Map.

4.2.4.1 Peripheral Map

The following table shows the address range of different aliased regions for each peripheral.

Address Range	Module Name
0x40000000 - 0x40003FFF	LETIMER0_NS
0x40004000 - 0x40007FFF	ADC0_NS
0x40008000 - 0x4000BFFF	ACMP0_NS
0x4000C000 - 0x4000FFFF	ACMP1_NS
0x40040000 - 0x40043FFF	PCNT0_NS
0x40100000 - 0x40103FFF	HFRCOEM23_NS
0x40104000 - 0x40107FFF	HFXO0_NS
0x40200000 - 0x40203FFF	I2C0_NS
0x40204000 - 0x40207FFF	WDOG0_NS
0x40208000 - 0x4020BFFF	EUSART0_NS
0x40800000 - 0x40803FFF	SOCPLL0_NS
0x40804000 - 0x40807FFF	CMU_NS
0x40808000 - 0x4080BFFF	DMEM_NS
0x40810000 - 0x40813FFF	HFRCO0_NS
0x40814000 - 0x40817FFF	LDMA0_NS
0x40818000 - 0x4081BFFF	FSRCO_NS
0x4081C000 - 0x4081FFFF	DPLL0_NS
0x40820000 - 0x40823FFF	LFXO_NS
0x40824000 - 0x40827FFF	LFRCO_NS
0x40828000 - 0x4082BFFF	ULFRCO_NS
0x4082C000 - 0x4082FFFF	SEMAPHORE0_NS
0x40830000 - 0x40833FFF	SEMAPHORE1_NS
0x40834000 - 0x40837FFF	L1ICACHE0_NS
0x40838000 - 0x4083BFFF	PRS_NS
0x4083C000 - 0x4083FFFF	GPIO_NS
0x40844000 - 0x40847FFF	LDMAXBAR0_NS
0x40848000 - 0x4084BFFF	TIMER0_NS
0x4084C000 - 0x4084FFFF	TIMER1_NS
0x40850000 - 0x40853FFF	TIMER2_NS
0x40854000 - 0x40857FFF	TIMER3_NS
0x40858000 - 0x4085BFFF	ETAMPDET_NS
0x4085C000 - 0x4085FFFF	L2ICACHE0_NS
0x40860000 - 0x40863FFF	PIXELRZ0_NS
0x40864000 - 0x40867FFF	BURTC_NS
0x40868000 - 0x4086BFFF	I2C1_NS
0x40870000 - 0x40873FFF	SYSCFG_NS
0x40874000 - 0x40877FFF	SYSCFG_NS
0x40878000 - 0x4087BFFF	BURAM_NS
0x4087C000 - 0x4087FFFF	GPCRC0_NS

Address Range	Module Name
0x40880000 - 0x40883FFF	LEDDRV0_NS
0x40884000 - 0x40887FFF	EUSART1_NS
0x40888000 - 0x4088BFFF	EUSART2_NS
0x4088C000 - 0x4088FFFF	PIXELRZ1_NS
0x40890000 - 0x40893FFF	I2C2_NS
0x40894000 - 0x40897FFF	WDOG1_NS
0x40898000 - 0x4089BFFF	RPA_NS
0x41000000 - 0x410007FF	SYMCRYPTO_NS
0x41004000 - 0x410047FF	AES_NS
0x41008000 - 0x4100BFFF	SMU_NS
0x4100C000 - 0x4100FFFF	SMU_NS
0x42800000 - 0x4280007F	SEMAILBOX_NS
0x50000000 - 0x50003FFF	LETIMER0
0x50004000 - 0x50007FFF	ADC0
0x50008000 - 0x5000BFFF	ACMP0
0x5000C000 - 0x5000FFFF	ACMP1
0x50040000 - 0x50043FFF	PCNT0
0x50100000 - 0x50103FFF	HFRCOEM23
0x50104000 - 0x50107FFF	HFXO0
0x50200000 - 0x50203FFF	I2C0
0x50204000 - 0x50207FFF	WDOG0
0x50208000 - 0x5020BFFF	EUSART0
0x50800000 - 0x50803FFF	SOCPLL0
0x50804000 - 0x50807FFF	CMU
0x50808000 - 0x5080BFFF	DMEM
0x50810000 - 0x50813FFF	HFRCO0
0x50814000 - 0x50817FFF	LDMA0
0x50818000 - 0x5081BFFF	FSRCO
0x5081C000 - 0x5081FFFF	DPLL0
0x50820000 - 0x50823FFF	LFXO
0x50824000 - 0x50827FFF	LFRCO
0x50828000 - 0x5082BFFF	ULFRCO
0x5082C000 - 0x5082FFFF	SEMAPHORE0
0x50830000 - 0x50833FFF	SEMAPHORE1
0x50834000 - 0x50837FFF	L1ICACHE0
0x50838000 - 0x5083BFFF	PRS
0x5083C000 - 0x5083FFFF	GPIO
0x50844000 - 0x50847FFF	LDMAXBAR0
0x50848000 - 0x5084BFFF	TIMER0
0x5084C000 - 0x5084FFFF	TIMER1
0x50850000 - 0x50853FFF	TIMER2

Address Range	Module Name
0x50854000 - 0x50857FFF	TIMER3
0x50858000 - 0x5085BFFF	ETAMPDET
0x5085C000 - 0x5085FFFF	L2ICACHE0
0x50860000 - 0x50863FFF	PIXELRZ0
0x50864000 - 0x50867FFF	BURTC
0x50868000 - 0x5086BFFF	I2C1
0x50870000 - 0x50873FFF	SYSCFG
0x50874000 - 0x50877FFF	SYSCFG
0x50878000 - 0x5087BFFF	BURAM
0x5087C000 - 0x5087FFFF	GPCRC0
0x50880000 - 0x50883FFF	LEDDRV0
0x50884000 - 0x50887FFF	EUSART1
0x50888000 - 0x5088BFFF	EUSART2
0x5088C000 - 0x5088FFFF	PIXELRZ1
0x50890000 - 0x50893FFF	I2C2
0x50894000 - 0x50897FFF	WDOG1
0x50898000 - 0x5089BFFF	RPA
0x51000000 - 0x510007FF	SYMCRYPTO
0x51004000 - 0x510047FF	AES
0x51008000 - 0x5100BFFF	SMU
0x5100C000 - 0x5100FFFF	SMU
0x52800000 - 0x5280007F	SEMAILBOX
0xA3000000 - 0xA3003FFF	SYSRTC0_NS
0xA3004000 - 0xA3007FFF	SCRATCHPAD_NS
0xA3008000 - 0xA300BFFF	EMU_NS
0xA300C000 - 0xA300FFFF	EMU_NS
0xA3010000 - 0xA3013FFF	SEPORTAL_NS
0xA3014000 - 0xA3017FFF	LPW0PORTAL_NS
0xA3018000 - 0xA301BFFF	HOSTPORTAL_NS
0xB3000000 - 0xB3003FFF	SYSRTC0
0xB3004000 - 0xB3007FFF	SCRATCHPAD
0xB3008000 - 0xB300BFFF	EMU
0xB300C000 - 0xB300FFFF	EMU
0xB3010000 - 0xB3013FFF	SEPORTAL
0xB3014000 - 0xB3017FFF	LPW0PORTAL
0xB3018000 - 0xB301BFFF	HOSTPORTAL

4.2.4.2 Peripheral Non-Word Access Behavior

When writing to peripheral registers, all accesses are treated as 32-bit accesses. This means that writes to a register need to be large enough to cover all bits of the register. Otherwise, any unwritten bits may become corrupted from the partial-word transfer. The safest practice is to always do 32-bit writes to peripheral registers.

When reading, there is generally no issue with partial word accesses. However, note that any read action (e.g. FIFO popping) will be triggered regardless of whether the actual FIFO bit-field was included in the transfer size.

4.2.4.3 Peripheral Bit Set and Clear

The SiXG301 supports bit set, bit clear, and bit toggle access to most peripheral registers. The bit set and bit clear functionality (also called "bit access") enables modification of bit fields without the need to perform a read-modify-write. Also, the operation is contained within a single bus access. Bit access registers and their addresses are shown in the register map for each peripheral. Peripherals without _SET, _CLR, or _TGL registers in the register map do not support these functions.

Each register with bit set functionality will have a _SET register. Whenever a bit in the SET register is written to a 1 the corresponding bit in its target register is set. The SET register is located at TARGET + 0x1000 where TARGET is the address of the target register and has the same name as the target register with '_SET' appended.

Each register with bit clear functionality will have a CLR register. Whenever a bit in the CLR register is written to a 1 the corresponding bit in its target register is cleared. The CLR register is located at TARGET + 0x2000 where TARGET is the address of the target register and has the same name as the target register with '_CLR' appended.

Each register with bit toggle functionality will have a TGL register. Whenever a bit in the TGL register is written to a 1 the corresponding bit in its target register is inverted. The TGL register is located at TARGET + 0x3000 where TARGET is the address of the target register and has the same name as the target register with '_TGL' appended.

Note: It is possible to use multiple bit clear and bit set operations in order to arbitrarily modify multi-bit register fields without affecting other fields in the same register. In this case, care should be taken to ensure that the field does not have intermediate values that can lead to erroneous behavior. For example, if a bit clear operation is followed by a bit set operation to change an analog tuning register field from 0x2 to 0x4 (first clearing bit 1 and then setting bit 2), the field would take on a value of zero for short time. If the analog module is active at the time, this could lead to undesired behavior.

4.2.4.4 Peripheral Access Performance

The Cortex-M33, LDMA Controller, and peripherals run on clocks which can be pre-scaled separately. Clocks and pre-scaling are described in more detail in 8. CMU - Clock Management Unit. This section describes the access performance for a peripheral register based on its frequency relative to the CPUCLK frequency and its access type. For this discussion, PERCLK refers to a selected peripheral's clock frequency and CPUCLK refers to a core's clock frequency.

The type of each register in a peripheral is indicated in the 'Access' column of the peripherals register table. Register types are: ENABLE, CONFIG, SYNC, LFSYNC, and INTFLAG. If not type is listed then the register is a Generic register.

4.2.4.4.1 Generic Registers

Registers with no type listed are generic registers. They may be read or written to at any time. Access will not stall the CPU.

4.2.4.4.2 CONFIG Registers

CONFIG Registers contain configuration that does not change during peripheral operation.

CONFIG registers may only be written when a peripheral is disabled. Writing to a CONFIG register when a peripheral is enabled will result in a BUSFAULT. CONFIG register writes will not stall the CPU.

CONFIG registers may be read at any time. Reads will not stall the CPU.

4.2.4.4.3 SYNC Registers

SYNC registers are used to communicate with running high-speed peripherals where PERCLK is expected to be within an order of magnitude of CPUCLK. For example a timer running at 76.8 MHz when the core is at 38.4 MHz or at 9.6 MHz when the core is 76.8 MHz. In this case CPU stalls of several PERCLK cycles do not significantly impact overall system performance in most systems.

SYNC registers may only be written to when the peripheral is enabled. Writing to a SYNC register when a peripheral is disabled will result in a BUSFAULT. A write will take several (2 - 3) PERCLK cycles to complete (take effect) during which time the entire module will be in a pending state. If a SYNC register is written to while the peripheral is already in a pending state, the CPU is stalled until the previous write finishes. If a SYNC register is written to while the peripheral is not in a pending state, the CPU is not stalled.

SYNC registers may be read at any time. If a SYNC register is read while the peripheral is disabled, the CPU is not stalled. If a SYNC register is read while the peripheral is enabled, the CPU will be stalled for several (2 -3) PERCLK cycles while up to date values are retrieved from the peripheral.

4.2.4.4.4 LFSYNC Registers

LFSYNC registers are used to communicate with running low frequency peripherals where PERCLK is expected to be much lower than the CPU clock and synchronization delays may be long. For example, a LETIMER running at 32 kHz when the core is at 76.8 MHz. In this case CPU stalls of several PERCLK cycles represent a significant blockage of the CPU and need to be avoided whenever possible. LFSYNC registers accommodate this by allowing the CPU to write the register and continue to do other work while the value is synchronized.

LFSYNC registers may be written at any time. A write will take several (3-4) PERCLK cycles to complete during which the register will be in a pending state. If a LFSYNC register is written to while it is in a pending state, a BUSFAULT will occur. Each LFSYNC register has a status bit indicating if it is currently pending.

LFSYNC registers may be read at any time. The CPU is never stalled on a read. If a LFSYNC register is read while pending, the pending (recently written) data will be returned even though it has not yet taken effect. Software may use the busy status bit to determine if the read value has been applied to the hardware.

4.2.4.4.5 ENABLE Registers

ENABLE registers contain the enable bit for a peripheral.

ENABLE registers may be written at any time. When the peripheral is enabled it takes some time for the enable to take effect during which time the module is pending. Peripherals will be in the pending state for a few (2-3) PERCLK cycles when first enabled. Since the clock source for the peripheral may not be running before the peripheral is enabled, the start up time for the clock source may increase the pending time. See 8. CMU - Clock Management Unit for more information on on-demand clock sources.

Disabling a high frequency module will stall the CPU until all pending SYNC writes have completed and any pending enable has completed. If the module is fully enabled and no SYNC writes are pending, the disable will be instantaneous. Disabling low frequency peripheral which a LFSYNC is pending will result in a bus fault. Disabling a low frequency peripherals while an enable is still pending causes no CPU stall.

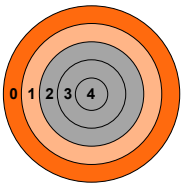
ENABLE registers may be read at any time.

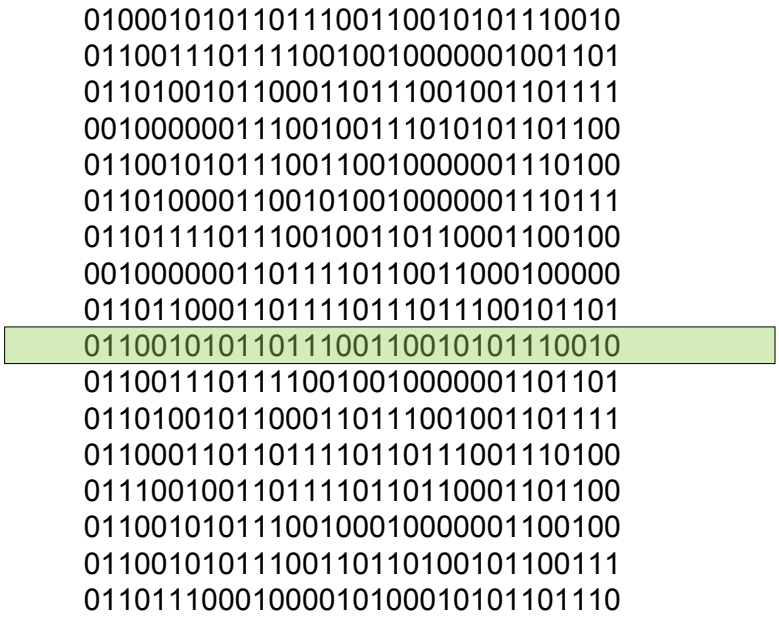
4.2.4.4.6 INTFLAG Registers

INTFLAG registers contain interrupt flags. To set or clear an interrupt flag, the _SET or _CLR register alias must be used. Writing directly to the INTFLAG register will have no effect.

Note that for an interrupt to occur when a flag is set the IRQ must be enabled in the NVIC.

5. System Memory and Configuration





Quick Facts

What?

The various system memories are accessible through dedicated hardware blocks.

Why?

Application code and user data can be stored in external flash memory as well as on-chip RAM. Dedicated hardware, including the EXTMEM subsystem, HYDRARAM controller, and ICACHE are use to ensure security and optimize performance.

How?

The EXTMEM subsystem handles access to flash memory, decryption, and authentication, while the HYDRARAM controller enables multi-port access to the on-chip DMEM RAM space.

EXTMEM includes a highly efficient local cache (L2CACHE) which reduces the number of flash reads significantly, improving performance and energy efficiency. The processor core's internal instruction cache (ICACHE) further optimizes the system. Built-in performance counters can be used to measure the efficiency of the instruction cache.

5.1 Introduction

Several system configuration and memory access blocks are available on the SiXG301, including:

- **EXTMEM** is a subsystem to access and control external serial flash memory, encompassing QSPI, FLPLL, FLREG, and L2CACHE.
- **HYDRARAM** is a RAM controller for accessing the on-chip DMEM space
- **ICACHE** is the host CPU core's internal instruction cache memory
- **DEVINFO** provides access to production-tested calibrations and other device-specific details
- **SYSCFG** is used to configure a small number of system-level resources such as the SYSTICK timer and bus fault exceptions

Refer to the following sections for more detail on each of these blocks.

5.2 EXTMEM - External Memory Interface

The External Memory interface (EXTMEM) controls off-chip program memory for the SiXG301 microcontroller.

5.2.1 Features

- QSPI serial flash interface with local PLL and flexible timing
- Integrated 1.8 V LDO (FLREG) to power external flash
- Built-in AES encryption/decryption and authentication hardware
- Partitioning for separate user code and data regions
  - Up to eight code regions can be designated as plaintext, encrypted, or authenticated and encrypted
  - Data region for storing nonvolatile application data
- Execute-in-place (XiP) and Authenticated execute-in-place (AXiP) support
- 64 KB 4-way set associative cache (L2CACHE)
- Flash read/write/erase management

5.2.2 Overview

EXTMEM interfaces to 1.8 V external flash memory through a high speed quad SPI interface (QSPI) with dedicated encryption, decryption and authentication hardware. An L2CACHE holds cached instruction data for efficient code execution. An integrated LDO powers the external memory at 1.8 V, removing the need for a second external supply in 3.3 V applications. The dedicated flash PLL (FLPLL) is used to optimize QSPI interface timing.

Flash memory contents are managed by the SE and the external memory hardware to provide robust security for program memory. The user can partition the flash memory into code and non-volatile data storage segments according to the needs of the application. Devices reserve 192 KB of the attached flash to store the upgradeable portion of SE firmware.

Code partitions can be stored using three different methods: unencrypted, encrypted, and authenticated. When storing a portion of code as authenticated, a portion of the partition is reserved to store authentication data (4 KB of authentication data for every 32 KB of code). The memory interface hardware manages addressing, decryption, and authentication. The interface hardware supports eXecute-in-Place (XiP) from all code partitions. Encrypted eXecute-in-Place (EXiP) of encrypted code regions does not require any additional overhead. Authenticated eXecute-in-Place (AXiP) requires a small amount of overhead to read the additional authentication signatures.

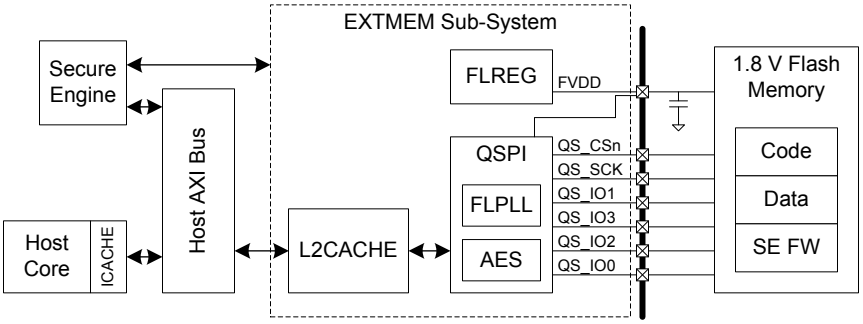


Figure 5.1. EXTMEM Overview

5.2.3 Flash Partitioning

The Secure Engine is used to partition external flash into up to eight user code partitions in the user code region and a user data region. Each code partition can be designated as plaintext, encrypted, or encrypted with authentication. The data region is a single contiguous area that is stored without encryption or authentication.

Physical space in the flash memory is allocated according to the region type and attributes. 192 KB is reserved for the SE firmware image. The user code partitions are allocated at different rates according to whether the code partition is designated for authentication. If a user code partition is using authentication, the partition will be allocated at 36 KB of physical space per 32 KB of logical address space (a 9:8 ratio). The extra 4 KB per 32 KB is used to accommodate authentication overhead. If a user code partition is stored as plaintext or encrypted (without authentication), the partition is allocated at 32 KB of phyiscal space per 32 KB of logical address space (a 1:1 ratio). The remainder of the physical space in the flash becomes the user data region.

The user code region starts at logical address 0x01000000, and the boundaries between code partitions can be designated at 32 KB logical addresses. The data region begins at the logical address after the final code partition. The EXTMEM interface handles all translation between logical addresses in the system and the physical addresses in the flash.

Figure 5.2. Flash Partitioning shows an example of a system with two authenticated partitions and one encrypted partition. This example assumes a 2 MB (2048 KB) flash.

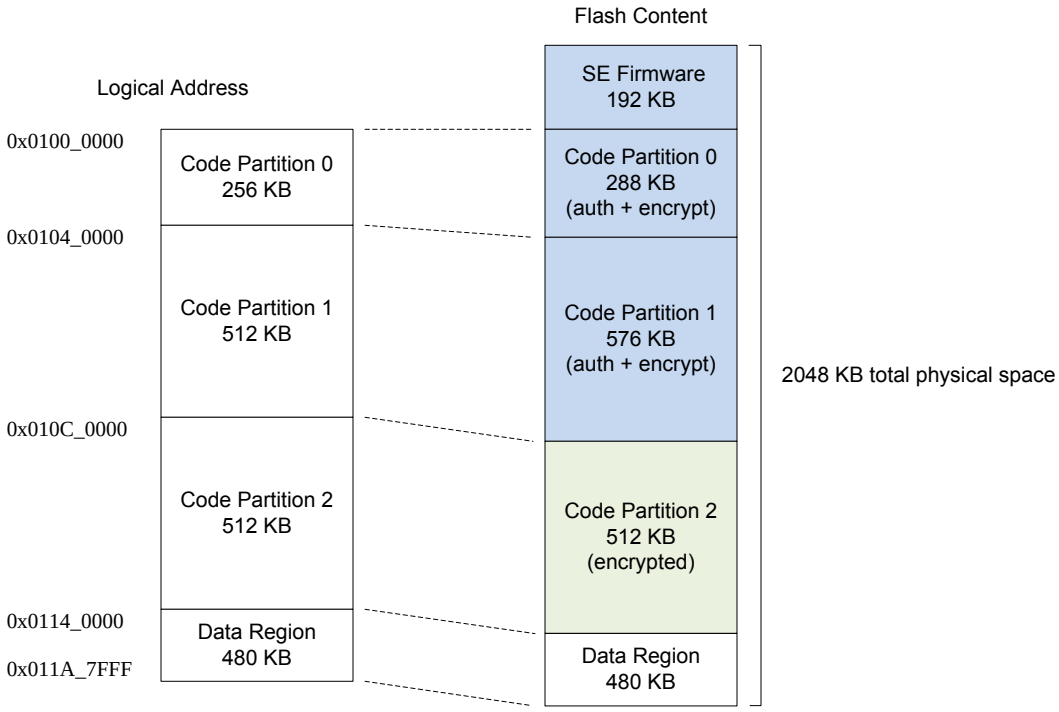


Figure 5.2. Flash Partitioning

In this example, code partition 0 is 256 KB with encryption and authentication, code partition 1 is 512 KB with encryption and authentication, and code partition 2 is 512 KB with encryption, for a total logical code region size of 1280 KB. Because code partition 0 and code partition 1 use authentication, these code partitions require an additional 4 KB of physical space for every 32 KB of code. Thus the total code region requires (256 + 32) KB + (512 + 64) KB + (512) KB = 1376 KB of storage space. The data region is placed after code space and uses the remainder of available flash, which is 2048 KB - 192 KB (SE firmware) - 1376 KB (code region) = 480 KB.

The logical addresses shown in Figure 5.2. Flash Partitioning are only one of four aliased locations where the memory can be accessed. The host CPU core has four aliases to handle the combination of Secure versus Non-Secure and Instruction (cacheable) versus Data (non-cacheable) data stored in EXTMEM. See Table 4.1. Host Core Address Map for more detail.

5.2.3.1 User Code Partitions

Code partitions are sized on 32 KB logical address boundaries, and may be erased in 32 KB chunks within that partition. Each partition may be designated to be stored and accessed without encryption, with encryption, or with authentication and encryption.

Partitions that are stored without encryption are written to the external flash as plaintext. If accessed from a cacheable address, the information for an entire line (32 bytes) is read and placed in the L2CACHE.

Partitions that are stored with encryption are encrypted with AES-CTR using a protected key with a partition and version-specific IV. Decryption is performed in-place by the EXTMEM hardware when reading from the external flash. If accessed from a cacheable address, the information for an entire line (32 bytes) is decrypted and placed in the L2CACHE.

Partitions that are stored with authentication and encryption are encrypted and authenticated with AES-GCM using a protected key with a partition and version-specific initialization vector (IV). The authentication information is included in the flash read and decrypted instructions are placed in the L2CACHE if accessed from a cacheable address.

Each code partition may be locked via the SE.

5.2.3.2 User Data Region

The data region is sized on 4 KB logical address boundaries, and may be erased in 4 KB chunks. The data in the data region is always stored as plaintext. This does not mean that encrypted data cannot be stored in the data region. It simply means that the EXTMEM interface does not encrypt or decrypt the data in the data region automatically. Software may still encrypt the data according to the needs of the application prior to writing the (encrypted) data to the flash.

5.2.4 Write and Erase

Writing and erasing the external flash is performed via the SE. External serial flash writes can take a long time. In many cases, code that needs to execute and respond quickly to events during a flash write or erase operation can be stored in L2CACHE or DMEM and executed from that memory area. In some applications this may not be feasible. The EXTMEM interface supports write and erase pausing, allowing an in-progress operation to be halted so that instructions can be fetched from flash, and then resumed at a later time. Write and erase pausing is managed by the SE. User firmware can indicate a maximum write time window to ensure that the system can respond in that time or sooner. A detected SE tamper event will also cause the SE to issue a pause operation. Note that instructions and data cannot be fetched from a sector that has a paused write or erase in progress.

5.2.5 Flash Regulator (FLREG)

The flash regulator (FLREG) is an on-chip LDO providing a 1.8V supply to the external or co-packaged flash memory, with the following features;

- 1.8 V output to power QSPI interface and external chip
- Low dropout allows input supply down to 1.9 V.
- Automatic power on/off to save energy in sleep states
- Can be disabled to allow external power source

The FLREG output routes to the FVDD pin, and must be bypassed with an appropriate capacitor as specified in the device data sheet. FVDD powers both the serial flash memory and the QSPI interface. In systems where an external 1.8 V supply is used to power the SiXG301, it is recommended to connect the external supply to FVDD and disable FLREG to save energy. FLREG is disabled by clearing the EMU_FLREG.EN bit to 0.

5.2.6 Quad SPI (QSPI)

The quad SPI interface is an integral part of the EXTMEM subsystem. It provides the physical interface to the external SPI flash device. The QSPI is fully managed by the SE hardware and provides the following features:

- Supports a wide variety of serial flash vendors with programmable timing and commands
- Support for single, dual, and quad-SPI operations
- Supports clock mode 0
- Support for SDR (to 160 MHz) and DDR mode (to 100 MHz)
- Execute-in-place (XiP) and Authenticated execute-in-place (AXiP) support
- Local PLL (FLPLL) and tunable delay lines to optimize performance and decouple operation from AXI bus clock

5.2.7 L2CACHE

The L2CACHE is a cache block in the EXTMEM subsystem, providing efficiency for instruction execution from the EXTMEM space. The L2CACHE is fully managed by the SE hardware and has the following high-level features:

- Multiple ports for up to 8 simultaneous reads
- 3-cycle hit latency
- 64 KB array
- 4-way set associative
- 256-bit cache lines

The L2CACHE serves any bus managers in the system that may access the EXTMEM space. Upon a read from EXTMEM, the L2CACHE quickly determines whether the read is already cached (hit) or requires a read from the external memory (miss). Cache hits return data in 3 SYSCLK cycles. Cache misses are forwarded to the QSPI hardware to retrieve the data from external flash memory. The QSPI hardware will perform any necessary read, decryption, and authentication to fill the entire 256-bit cache line.

5.3 HYDRARAM - RAM Controller

The HYDRARAM block is a multi-port, multi-protocol RAM controller that allows simultaneous access to different RAM banks in DMEM space. RAM controller port priorities, ECC, and other RAM controls are available.

5.3.1 Features

The HYDRARAM features include:

- Support for concurrent access to all ports
- ECC with correction of 1-bit errors and detection of 2-bit errors
- Interrupt generation for ECC errors
- Dynamic configuration of memory banks as interleaved or sequential
- Read caching for sequential access
- Priority rank arbitration
- Global monitor support

5.3.2 RAM Bank Caching

Sequential reads from DMEM banks can optionally be cached (buffered) to improve timing and reduce overall energy. Caching is enabled on a per-bank basis using the BANKSCACHEEN register. Each bit of BANKSCACHEEN enables caching for one RAM bank.

When caching is enabled, a read will buffer four 32-bit words aligned on a 4-word boundary and containing the read address. Reading sequential memory locations will result in lower power if caching is enabled. Conversely, reading random locations may increase power if caching is enabled. Caching is not recommended for interleaved DMEM regions.

5.3.3 RAM Bank Interleaving

The HYDRARAM controller allows regions of DMEM to be interleaved in either 2, 4, or 8-bank groups. Interleaving memory banks may be advantageous when multiple managers may access the same memory region simultaneously. If there is only one manager accessing a memory region, then there is no performance gain for interleaving. Moreover, if multiple managers each would access a separate memory region that do not share the same memory banks, then it is more advantageous to leave that memory region as sequential.

By default, all RAM in DMEM space is addressed sequentially. For example, reading from address 0x0 in DMEM space will access the 32-bit word at address 0x0 in Bank 0. Reading from address 0x4 will access the 32-bit word at address 0x4 in Bank 0.

When memory banks are interleaved, adjacent addresses on 32-bit boundaries access different RAM banks. For example, if 2-bank interleaving is used with bank 0 and bank 1, address 0x0 maps to address 0x0 in Bank 0, address 0x4 maps to address 0x0 in Bank 1, address 0x8 maps to address 0x4 in Bank 0, and so on. This is illustrated further in Table 5.1. Interleaved Access.

Table 5.1. Interleaved Access

Logical Address in DMEM	2-Bank Interleave	4-Bank Interleave	8-Bank Interleave
0x0	Bank 0, addr 0x0	Bank 0, addr 0x0	Bank 0, addr 0x0
0x4	Bank 1, addr 0x0	Bank 1, addr 0x0	Bank 1, addr 0x0
0x8	Bank 0, addr 0x4	Bank 2, addr 0x0	Bank 2, addr 0x0
0xC	Bank 1, addr 0x4	Bank 3, addr 0x0	Bank 3, addr 0x0
0x10	Bank 0, addr 0x8	Bank 0, addr 0x4	Bank 4, addr 0x0
0x14	Bank 1, addr 0x8	Bank 1, addr 0x4	Bank 5, addr 0x0
0x18	Bank 0, addr 0xC	Bank 2, addr 0x4	Bank 6, addr 0x0
0x1C	Bank 1, addr 0xC	Bank 3, addr 0x4	Bank 7, addr 0x0

Interleaving is controlled by the INTERLEAVE2BANK, INTERLEAVE4BANK, and INTERLEAVE8BANK registers in HYDRARAM. Each bit in these registers controls interleaving for the specified number of banks, starting with bank 0 in the memory map. Setting bit 0 of INTERLEAVE2BANK will configure a 2-bank region consisting of Bank 0 and Bank 1 for interleave. Setting bit 1 of INTERLEAVE2BANK will configure the 2-bank region consisting of Bank 2 and Bank 3 for interleave. The other two control registers work in the same way, where bit 0 of INTERLEAVE4BANK configures Banks 0, through 3 for 4-bank interleave, and bit 0 of INTERLEAVE8BANK configures Banks 0, through 7 for 8-bank interleave.

Note: No bank in DMEM should be configured for more than one interleave option, and it is up to software to ensure valid configuration. Configuring DMEM regions for interleave should be performed before that region is used to store data.

5.3.4 Bus Address Faults

By default, reads to invalid accesses in DMEM space will trigger a bus fault. The ADDRFAULTEN bit in HYDRARAM_CTRL may be used to disable fault generation on these invalid access events.

5.3.5 ECC

DMEM0 supports one bit correction and two bit detection ECC.

To enable error detection for DMEM0, set ECCEN in HYDRARAM_CTRL. To enable auto-correction of one bit errors in DMEM0, set ECCWEN in HYDRARAM_CTRL.

When ECC error events in DMEM0 are detected, the corresponding bits in HYDRARAM_IF are set. When a flag is set, an interrupt will be triggered if the corresponding interrupt enable bit is set in HYDRARAM_IEN. When an error occurs, the address of the detected error is written to HYDRARAM_ECCERRADDR. The ECCERRADDR register will contain the address of the most recently detected error. It can be cleared using HYDRARAM_CMD.CLEARECC. If multiple ECC errors occur without ECCERRADDR being cleared, the MERRIND bit in HYDRARAM_ECCMERRIND will be set. This status bit is sticky, and is also cleared with HYDRARAM_CMD.CLEAR-ECC.

Upon a two bit ECC error in DMEM, HYDRARAM can also issue a bus fault. To enable this, set the ECCERRFAULTEN bit in HYDRARAM_CTRL.

The recommend procedure for initializing ECC RAM is to first enable ECC, then write zeros to all locations. This will clear the RAM and initialize the syndrome. If the ECC RAM is not written as described, then any reads to uninitialized RAM locations will result in an ECC error.

Note: The RAM ECC feature must be enabled to achieve good long term reliability. The long term reliability of the RAM is only specified with ECC enabled.

5.3.6 Port Arbitration Priority

Different ports can access different banks of DMEM simultaneously. When two ports try to access the same DMEM bank at the same time however, the HYDRARAM controller arbitrates the access based on a priority scheme. The PORTPRIORITY register has a PORTPRIORITYx field for each HYDRARAM port. PORTPRIORTYx allows software to increase the priority given to one or more ports. Arbitration gives priority to the ports with the highest PORTPRIORITYx values, and when PORTPRIORITYx is the same for multiple ports, priority is given to ports with the highest index.

For example, considering four ports (PORT0, PORT1, PORT2, and PORT3), if the PORTPRIORITY1 field is set to 1 and all other PORTPRIORTYx fields are 0, then the priority order of the four ports (highest to lowest) is PORT1, PORT3, PORT2, PORT0.

5.3.7 HYDRARAM Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	HYDRARAM_IPVERSION	R	IP Version ID
0x004	HYDRARAM_CMD	W	Command Register
0x008	HYDRARAM_CTRL	RW	Control Register
0x00C	HYDRARAM_PORTPRIORITY	RW	Port Arbitration Priority Register
0x010	HYDRARAM_INTER-LEAVE2BANK	RW	2 Bank Interleaving Register
0x014	HYDRARAM_INTER-LEAVE4BANK	RW	4 Bank Interleaving Register
0x018	HYDRARAM_INTER-LEAVE8BANK	RW	8 Bank Interleaving Register
0x01C	HYDRARAM_ECCERRADDR	RH	ECC Error Address Register
0x020	HYDRARAM_ECCMERRIND	RH	Multiple ECC Error Indication
0x024	HYDRARAM_IF	RWH INTFLAG	Interrupt Flags
0x028	HYDRARAM_IEN	RW	Interrupt Enable
0x02C	HYDRARAM_BANKSCACHEEN	RW	RAM Bank Cache Enable
0x1000	HYDRARAM_IPVERSION_SET	R	IP Version ID
0x1004	HYDRARAM_CMD_SET	W	Command Register
0x1008	HYDRARAM_CTRL_SET	RW	Control Register
0x100C	HYDRARAM_PORTPRIORI-TY_SET	RW	Port Arbitration Priority Register
0x1010	HYDRARAM_INTER-LEAVE2BANK_SET	RW	2 Bank Interleaving Register
0x1014	HYDRARAM_INTER-LEAVE4BANK_SET	RW	4 Bank Interleaving Register
0x1018	HYDRARAM_INTER-LEAVE8BANK_SET	RW	8 Bank Interleaving Register
0x101C	HYDRARAM_ECCER-RADDR_SET	RH	ECC Error Address Register
0x1020	HYDRARAM_ECCMER-RIND_SET	RH	Multiple ECC Error Indication
0x1024	HYDRARAM_IF_SET	RWH INTFLAG	Interrupt Flags
0x1028	HYDRARAM_IEN_SET	RW	Interrupt Enable
0x102C	HYDRARAM_BANKSCAC-HEEN_SET	RW	RAM Bank Cache Enable
0x2000	HYDRARAM_IPVERSION_CLR	R	IP Version ID
0x2004	HYDRARAM_CMD_CLR	W	Command Register
0x2008	HYDRARAM_CTRL_CLR	RW	Control Register
0x200C	HYDRARAM_PORTPRIORI-TY_CLR	RW	Port Arbitration Priority Register
0x2010	HYDRARAM_INTER-LEAVE2BANK_CLR	RW	2 Bank Interleaving Register
0x2014	HYDRARAM_INTER-LEAVE4BANK_CLR	RW	4 Bank Interleaving Register

Offset	Name	Type	Description
0x2018	HYDRARAM_INTERLEAVE8BANK_CLR	RW	8 Bank Interleaving Register
0x201C	HYDRARAM_ECCERRADDR_CLR	RH	ECC Error Address Register
0x2020	HYDRARAM_ECCMERRIND_CLR	RH	Multiple ECC Error Indication
0x2024	HYDRARAM_IF_CLR	RWH INTFLAG	Interrupt Flags
0x2028	HYDRARAM_IEN_CLR	RW	Interrupt Enable
0x202C	HYDRARAM_BANKSCACHEEN_CLR	RW	RAM Bank Cache Enable
0x3000	HYDRARAM_IPVERSION_TGL	R	IP Version ID
0x3004	HYDRARAM_CMD_TGL	W	Command Register
0x3008	HYDRARAM_CTRL_TGL	RW	Control Register
0x300C	HYDRARAM_PORTPRIORITY_TGL	RW	Port Arbitration Priority Register
0x3010	HYDRARAM_INTERLEAVE2BANK_TGL	RW	2 Bank Interleaving Register
0x3014	HYDRARAM_INTERLEAVE4BANK_TGL	RW	4 Bank Interleaving Register
0x3018	HYDRARAM_INTERLEAVE8BANK_TGL	RW	8 Bank Interleaving Register
0x301C	HYDRARAM_ECCERRADDR_TGL	RH	ECC Error Address Register
0x3020	HYDRARAM_ECCMERRIND_TGL	RH	Multiple ECC Error Indication
0x3024	HYDRARAM_IF_TGL	RWH INTFLAG	Interrupt Flags
0x3028	HYDRARAM_IEN_TGL	RW	Interrupt Enable
0x302C	HYDRARAM_BANKSCACHEEN_TGL	RW	RAM Bank Cache Enable

5.3.8 HYDRARAM Register Description

5.3.8.1 HYDRARAM_IPVERSION - IP Version ID

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															R	
Name																															IPVERSION	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	IPVERSION	0x1	R	**IP Version ID**
	IP Version Number			

5.3.8.2 HYDRARAM_CMD - Command Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																W(nB)
Name																																CLEARECC

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CLEARECC	0x0	W(nB)	**Clear ECCERRADDR and ECCMERRIND.MERRIND**
	Write 1 to clear ECCERRADDR and ECCMERRIND.MERRIND			

5.3.8.3 HYDRARAM_CTRL - Control Register

Offset	Bit Position																															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x1	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													ADDRFAULTEN	ECCERRFAULTEN	ECCWEN	ECCEN

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	ADDRFAULTEN	0x1	RW	**Address fault bus fault enable**
	When this bit is set, bus faults are generated on accesses to invalid parts of the HYDRARAM memory space.			
2	ECCERRFAULTEN	0x0	RW	**ECC Error bus fault enable**
	Enable address bus faults resulting from ECC errors			
1	ECCWEN	0x0	RW	**Enable ECC syndrome writes**
	Enable ECC syndrome writes			
0	ECCEN	0x0	RW	**Enable ECC functionality**
	Enable ECC functionality			

5.3.8.4 HYDRARAM_PORTPRIORITY - Port Arbitration Priority Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																										0x0		0x0		0x0		0x0
Access																										RW		RW		RW		RW
Name																										PORTPRIORITY3		PORTPRIORITY2		PORTPRIORITY1		PORTPRIORITY0

Bit	Name	Reset	Access	Description
31:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	PORTPRIORITY3	0x0	RW	**Port3 Arbitration Priority**
	Configure Arbitration Priority for HYDRARAM port3.			
5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PORTPRIORITY2	0x0	RW	**Port2 Arbitration Priority**
	Configure Arbitration Priority for HYDRARAM port2.			
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	PORTPRIORITY1	0x0	RW	**Port1 Arbitration Priority**
	Configure Arbitration Priority for HYDRARAM port1.			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	PORTPRIORITY0	0x0	RW	**Port0 Arbitration Priority**
	Configure Arbitration Priority for HYDRARAM port0.			

5.3.8.5 HYDRARAM_INTERLEAVE2BANK - 2 Bank Interleaving Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																	INTL2BANKBIT15	INTL2BANKBIT14	INTL2BANKBIT13	INTL2BANKBIT12	INTL2BANKBIT11	INTL2BANKBIT10	INTL2BANKBIT9	INTL2BANKBIT8	INTL2BANKBIT7	INTL2BANKBIT6	INTL2BANKBIT5	INTL2BANKBIT4	INTL2BANKBIT3	INTL2BANKBIT2	INTL2BANKBIT1	INTL2BANKBIT0

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15	INTL2BANKBIT15	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*15 and 2*15+1			
14	INTL2BANKBIT14	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*14 and 2*14+1			
13	INTL2BANKBIT13	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*13 and 2*13+1			
12	INTL2BANKBIT12	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*12 and 2*12+1			
11	INTL2BANKBIT11	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*11 and 2*11+1			
10	INTL2BANKBIT10	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*10 and 2*10+1			
9	INTL2BANKBIT9	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*9 and 2*9+1			
8	INTL2BANKBIT8	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*8 and 2*8+1			
7	INTL2BANKBIT7	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*7 and 2*7+1			
6	INTL2BANKBIT6	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*6 and 2*6+1			
5	INTL2BANKBIT5	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*5 and 2*5+1			
4	INTL2BANKBIT4	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*4 and 2*4+1			
3	INTL2BANKBIT3	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*3 and 2*3+1			
2	INTL2BANKBIT2	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*2 and 2*2+1			

Bit	Name	Reset	Access	Description
1	INTL2BANKBIT1	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*1 and 2*1+1			
0	INTL2BANKBIT0	0x0	RW	**Per 2 bank interleaving bit**
	Bit for two bank interleaving for banks 2*0 and 2*0+1			

5.3.8.6 HYDRARAM_INTERLEAVE4BANK - 4 Bank Interleaving Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																									RW	RW	RW	RW	RW	RW	RW	RW
Name																									INTL4BANKBIT7	INTL4BANKBIT6	INTL4BANKBIT5	INTL4BANKBIT4	INTL4BANKBIT3	INTL4BANKBIT2	INTL4BANKBIT1	INTL4BANKBIT0

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	INTL4BANKBIT7	0x0	RW	**Per 4 bank interleaving bit**
	Bit for four bank interleaving for banks 4*7,4*7+1,4*7+2 and 4*7+3			
6	INTL4BANKBIT6	0x0	RW	**Per 4 bank interleaving bit**
	Bit for four bank interleaving for banks 4*6,4*6+1,4*6+2 and 4*6+3			
5	INTL4BANKBIT5	0x0	RW	**Per 4 bank interleaving bit**
	Bit for four bank interleaving for banks 4*5,4*5+1,4*5+2 and 4*5+3			
4	INTL4BANKBIT4	0x0	RW	**Per 4 bank interleaving bit**
	Bit for four bank interleaving for banks 4*4,4*4+1,4*4+2 and 4*4+3			
3	INTL4BANKBIT3	0x0	RW	**Per 4 bank interleaving bit**
	Bit for four bank interleaving for banks 4*3,4*3+1,4*3+2 and 4*3+3			
2	INTL4BANKBIT2	0x0	RW	**Per 4 bank interleaving bit**
	Bit for four bank interleaving for banks 4*2,4*2+1,4*2+2 and 4*2+3			
1	INTL4BANKBIT1	0x0	RW	**Per 4 bank interleaving bit**
	Bit for four bank interleaving for banks 4*1,4*1+1,4*1+2 and 4*1+3			
0	INTL4BANKBIT0	0x0	RW	**Per 4 bank interleaving bit**
	Bit for four bank interleaving for banks 4*0,4*0+1,4*0+2 and 4*0+3			

5.3.8.7 HYDRARAM_INTERLEAVE8BANK - 8 Bank Interleaving Register

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													INTL8BANKBIT3	INTL8BANKBIT2	INTL8BANKBIT1	INTL8BANKBIT0

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	INTL8BANKBIT3	0x0	RW	**Per 8 bank interleaving bit**
	Bit for eight bank interleaving for banks 8*3,8*3+1,8*3+2, 8*3+3, 8*3+4, 8*3+5, 8*3+6, 8*3+7			
2	INTL8BANKBIT2	0x0	RW	**Per 8 bank interleaving bit**
	Bit for eight bank interleaving for banks 8*2,8*2+1,8*2+2, 8*2+3, 8*2+4, 8*2+5, 8*2+6, 8*2+7			
1	INTL8BANKBIT1	0x0	RW	**Per 8 bank interleaving bit**
	Bit for eight bank interleaving for banks 8*1,8*1+1,8*1+2, 8*1+3, 8*1+4, 8*1+5, 8*1+6, 8*1+7			
0	INTL8BANKBIT0	0x0	RW	**Per 8 bank interleaving bit**
	Bit for eight bank interleaving for banks 8*0,8*0+1,8*0+2, 8*0+3, 8*0+4, 8*0+5, 8*0+6, 8*0+7			

5.3.8.8 HYDRARAM_ECCERRADDR - ECC Error Address Register

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	ADDR																															

Bit	Name	Reset	Access	Description
31:0	ADDR	0x0	R	**ECC Error Address**
	Last captured ECC error address for the HYDRARAM. Cleared by CMD.CLEARECC			

5.3.8.9 HYDRARAM_ECCMERRIND - Multiple ECC Error Indication

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																MERRIND

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	MERRIND	0x0	R	**Multiple ECC errors on HYDRARAM**
	Multiple ECC error indication for HYDRARAM. Cleared by CMD.CLEARECC.			

5.3.8.10 HYDRARAM_IF - Interrupt Flags

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														INTEGRITYERR	ERR2B	ERR1B

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	INTEGRITYERR	0x0	RW	**Integrity check error Interrupt flag**
	Incoming integrity check error interrupt flag			
1	ERR2B	0x0	RW	**2-bit ECC Error Interrupt Flag**
	2-bit ECC Error Interrupt Flag for HYDRARAM			
0	ERR1B	0x0	RW	**1-bit ECC Error Interrupt Flag**
	1-bit ECC Error Interrupt Flag for HYDRARAM			

5.3.8.11 HYDRARAM_IEN - Interrupt Enable

Offset	Bit Position																															
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														INTEGRITYERR	ERR2B	ERR1B

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	INTEGRITYERR	0x0	RW	**Integrity check error Interrupt Enable**
	Incoming integrity check error interrupt Enable			
1	ERR2B	0x0	RW	**2-bit ECC Error Interrupt Enable**
	2-bit ECC Error Interrupt Enable for HYDRARAM			
0	ERR1B	0x0	RW	**1-bit ECC Error Interrupt Enable**
	1-bit ECC Error Interrupt Enable for HYDRARAM			

5.3.8.12 HYDRARAM_BANKSCACHEEN - RAM Bank Cache Enable

Offset	Bit Position																															
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	BANKSCACHEEN																															

Bit	Name	Reset	Access	Description
31:0	BANKSCACHEEN	0x0	RW	**Enable bit for RAM bank caching**
	Each bit controls if a bank enables internal RAM bank cache			

5.4 ICACHE - Instruction Cache

The ICACHE provides fast access to recently executed instructions, improving both speed and power consumption of code execution. The instruction cache is enabled by default, but can be disabled by setting CACHEDIS in ICACHE_CTRL. When enabled, the instruction cache typically reduces the number of flash reads significantly, thus saving energy. In most cases, a cache hit-rate of more than 70 % is achievable. When a 32-bit instruction fetch hits in the cache, the data is returned to the processor in one clock cycle, bypassing the flash access wait-states.

The instruction cache is connected directly to the CODE bus on the ARM core and functions as a memory access filter between the processor and the memory system, as illustrated in Figure 5.3. Instruction Cache Block Diagram. The cache consists of an access filter, lookup logic, SRAM, and three performance counters. The access filter checks if a transfer is an instruction fetch located in a cacheable region. If it is the cache lookup logic and SRAM is enabled. Otherwise, the cache is bypassed and the access is forwarded to the memory system. If lookup is enabled data is either returned from the cache (hit) or fetch from the memory system and cached (miss).

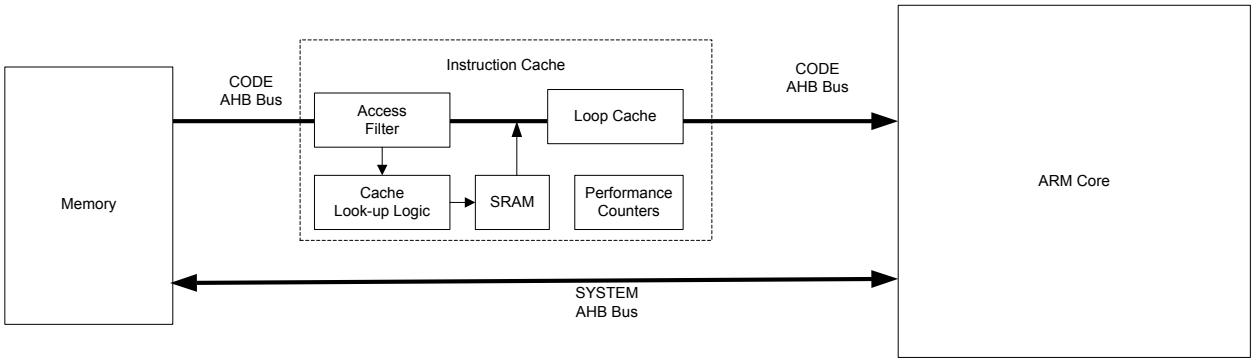


Figure 5.3. Instruction Cache Block Diagram

Note that while all access to code spaces use the CODE bus only instruction fetches are cached. Data accesses to the CODE region are passed through the ICACHE.

5.4.1 Cache Operation

It is highly recommended to keep the cache enabled. To improve cache-efficiency, sections of code with very low cache hit rate should not be cached. This is achieved by placing these code sections in non-cacheable MPU regions. This feature may also be used to avoid instructions from low-power memory taking up space from more power-hungry memory. For more information on the MPU see the ARM Cortex-M33 MPU documentation.

The cache can also be manually invalidated by writing 1 to INVALIDATE in ICACHE_CMD.

In the event that a parity error in the cache is detected, the RAMERROR flag will be set in ICACHE_IF. The data is automatically reloaded when this occurs so no action is required by software. This flag is informational only and can be used to detect the rate of corruption events. If RAMERROR in ICACHE_IEN is set, an interrupt will be triggered.

The cache line being filled is automatically invalidated whenever a bus fault occurs.

If the downstream memory reports a bus-fault at any time during a cache refill, the ICACHE invalidates the cache line that is being refilled to make sure it does not store any corrupted values. A bus-fault will pass through the ICACHE back to the core for the first read response data on a cache refill, so no further actions are taken. If a bus fault occurs outside of a cache refill, it will always be passed through the ICACHE back to the core.

5.4.2 Performance Measurement

To measure the hit-rate of a code-section, the built-in performance counters can be used. Before the section, start the performance counters by setting STARTPC in ICACHE_CMD register. This starts the performance counters, counting from 0. At the end of the section, stop the performance counters by setting STOPPC in ICACHE_CMD. The number of cache hits and cache misses for that section can then be read from PCHITS and PCMISSES. The cache hit-ratio can be calculated as PCHITS / (PCHITS + PCMISSES). PCAHITS contains the loopcache hits only. Any hits in PCAHITS are also counted in PCHITS. The loopcache hit-ratio can be calculated as PCAHITS / (PCHITS + PCMISSES). When PCHITS/PCAHITS/PCMISSES overflow, the HITOF/AHITOF/MISSOF interrupt flags are set respectively. These flags must be cleared by software. The range of the performance counters can be extended by increasing a counter in the interrupt routine. The performance counters only count when a cache lookup is performed. Access to non-cacheable regions, data fetches, and access made while the ICACHE is disabled do not increment PCMISSES.

Software may check the if the performance counters are running using PCRUNNING in ICACHE_STATUS.

5.4.3 ICACHE Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	ICACHE_IPVERSION	R	IP Version
0x004	ICACHE_CTRL	RW	Control Register
0x008	ICACHE_PCHITS	RH	Performance Counter Hits
0x00C	ICACHE_PCMISSES	RH	Performance Counter Misses
0x010	ICACHE_PCAHITS	RH	Performance Counter Advanced Hits
0x014	ICACHE_STATUS	RH	Status Register
0x018	ICACHE_CMD	W	Command Register
0x01C	ICACHE_LPMODE	RW	Low Power Mode
0x020	ICACHE_IF	RWH INTFLAG	Interrupt Flag
0x024	ICACHE_IEN	RW	Interrupt Enable
0x1000	ICACHE_IPVERSION_SET	R	IP Version
0x1004	ICACHE_CTRL_SET	RW	Control Register
0x1008	ICACHE_PCHITS_SET	RH	Performance Counter Hits
0x100C	ICACHE_PCMISSES_SET	RH	Performance Counter Misses
0x1010	ICACHE_PCAHITS_SET	RH	Performance Counter Advanced Hits
0x1014	ICACHE_STATUS_SET	RH	Status Register
0x1018	ICACHE_CMD_SET	W	Command Register
0x101C	ICACHE_LPMODE_SET	RW	Low Power Mode
0x1020	ICACHE_IF_SET	RWH INTFLAG	Interrupt Flag
0x1024	ICACHE_IEN_SET	RW	Interrupt Enable
0x2000	ICACHE_IPVERSION_CLR	R	IP Version
0x2004	ICACHE_CTRL_CLR	RW	Control Register
0x2008	ICACHE_PCHITS_CLR	RH	Performance Counter Hits
0x200C	ICACHE_PCMISSES_CLR	RH	Performance Counter Misses
0x2010	ICACHE_PCAHITS_CLR	RH	Performance Counter Advanced Hits
0x2014	ICACHE_STATUS_CLR	RH	Status Register
0x2018	ICACHE_CMD_CLR	W	Command Register
0x201C	ICACHE_LPMODE_CLR	RW	Low Power Mode
0x2020	ICACHE_IF_CLR	RWH INTFLAG	Interrupt Flag
0x2024	ICACHE_IEN_CLR	RW	Interrupt Enable
0x3000	ICACHE_IPVERSION_TGL	R	IP Version
0x3004	ICACHE_CTRL_TGL	RW	Control Register
0x3008	ICACHE_PCHITS_TGL	RH	Performance Counter Hits
0x300C	ICACHE_PCMISSES_TGL	RH	Performance Counter Misses
0x3010	ICACHE_PCAHITS_TGL	RH	Performance Counter Advanced Hits
0x3014	ICACHE_STATUS_TGL	RH	Status Register
0x3018	ICACHE_CMD_TGL	W	Command Register
0x301C	ICACHE_LPMODE_TGL	RW	Low Power Mode

Offset	Name	Type	Description
0x3020	ICACHE_IF_TGL	RWH INTFLAG	Interrupt Flag
0x3024	ICACHE_IEN_TGL	RW	Interrupt Enable

5.4.4 ICACHE Register Description

5.4.4.1 ICACHE_IPVERSION - IP Version

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

5.4.4.2 ICACHE_CTRL - Control Register

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0			0x0
Access																												RW	RW			RW
Name																												DATACACHEDIS	RAMRETAINDIS			CACHEDIS

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	DATACACHEDIS	0x0	RW	**Data cache disable**
	When set, cacheable data fetch will not be stored in cache RAM			
3	RAMRETAINDIS	0x0	RW	**RAM retain disable**
	When set, Cache RAM fetch will not be retained during sleep			
2:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CACHEDIS	0x0	RW	**Cache Disable**
	Disables caching			

5.4.4.3 ICACHE_PCHITS - Performance Counter Hits

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	PCHITS																															

Bit	Name	Reset	Access	Description
31:0	PCHITS	0x0	R	**Performance Counter Hits**
	Hit counter value			

5.4.4.4 ICACHE_PCMISSES - Performance Counter Misses

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	PCMISSES																															

Bit	Name	Reset	Access	Description
31:0	PCMISSES	0x0	R	**Performance Counter Misses**
	Miss counter value			

5.4.4.5 ICACHE_PCAHITS - Performance Counter Advanced Hits

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	PCAHITS																															

Bit	Name	Reset	Access	Description
31:0	PCAHITS	0x0	R	**Performance Counter Advanced Hits**
	Hit counter value for hits due to Advanced Buffering mode. These hits are also represented in PCHITS.			

5.4.4.6 ICACHE_STATUS - Status Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																PCRUNNING

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	PCRUNNING	0x0	R	**PC Running**
	Performance Counters are running			

5.4.4.7 ICACHE_CMD - Command Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														W(nB)	W(nB)	W(nB)
Name																														STOPPC	STARTPC	INVALIDATE

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	STOPPC	0x0	W(nB)	**Stop Performance Counters**
	Stops the Performance Counters			
1	STARTPC	0x0	W(nB)	**Start Performance Counters**
	Starts the Performance Counters			
0	INVALIDATE	0x0	W(nB)	**Invalidate**
	Invalidate the cached data			

5.4.4.8 ICACHE_LPMODE - Low Power Mode

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x2						0x3	
Access																									RW						RW	
Name																									NESTFACTOR						LPLEVEL	

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:4	NESTFACTOR	0x2	RW	**Low Power Nest Factor** Parameter used in the advanced buffering mode to control its estimation when a branch access is likely to be accessed in the near future. In general, a higher number will improve performance in code with deeply nested loops.
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	LPLEVEL	0x3	RW	**Low Power Level** Controls the low-power level of the cache. In general, the default setting is best for most applications.

Value	Mode	Description
0	BASIC	Base instruction cache functionality
1	ADVANCED	Advanced buffering mode, where the cache uses the fetch pattern to predict highly accessed data and store it in low-energy memory
3	MINACTIVITY	Minimum activity mode, which allows the cache to minimize activity in logic that it predicts has a low probability being used. This mode can introduce wait-states into the instruction fetch stream when the cache exits one of its low-activity states. The number of wait-states introduced is small, but users running with 0-wait-state memory and wishing to reduce the variability that the cache might introduce with additional wait-states may wish to lower the cache low-power level. Note, this mode includes the advanced buffering mode functionality.

5.4.4.9 ICACHE_IF - Interrupt Flag

Offset	Bit Position																														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0						0x0	0x0	0x0
Access																								RW						RW	RW	RW
Name																								RAMERROR						AHITOF	MISSOF	HITOF

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	RAMERROR	0x0	RW	**RAM error Interrupt Flag**
	RAM parity error detected			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	AHITOF	0x0	RW	**Advanced Hit Overflow Interrupt Flag**
	Advanced hit performance counter has overflowed			
1	MISSOF	0x0	RW	**Miss Overflow Interrupt Flag**
	Miss performance counter has overflowed			
0	HITOF	0x0	RW	**Hit Overflow Interrupt Flag**
	Hit performance counter has overflowed			

5.4.4.10 ICACHE_IEN - Interrupt Enable

Offset	Bit Position																														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0						0x0	0x0	0x0
Access																								RW						RW	RW	RW
Name																								RAMERROR						AHITOF	MISSOF	HITOF

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	RAMERROR	0x0	RW	**RAM error Interrupt Enable**
	Enable RAMERROR interrupt			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	AHITOF	0x0	RW	**Advanced Hit Overflow Interrupt Enable**
	Enable AHITOF interrupt			
1	MISSOF	0x0	RW	**Miss Overflow Interrupt Enable**
	Enable MISSOF interrupt			
0	HITOF	0x0	RW	**Hit Overflow Interrupt Enable**
	Enable HITOF interrupt			

5.5 Device Information

5.5.1 Introduction

The Device Information space is a nonvolatile memory space that is programmed with device information, including part number, calibration settings, and device feature information.

5.5.2 Description

The Device Information space consists of several groups of data, referred to as fragments. The available fragments are detailed in 5.5.4 Device Info Fragments. The device information space is only directly accessible through the Secure Engine. All fragments may be read using the SE Manager API via the ReadDeviceData mailbox command. All or part of each fragment may be read and stored in RAM data structures for future access.

5.5.3 DEVINFO Register Block

For convenience, the BASE fragment is also available in a set of hardware registers in the DEVINFO block beginning at address 0x0FE08100. The BASE fragment values can be directly accessed at that location without making an SE request.

5.5.4 Device Info Fragments

Table 5.2. BASE Fragment

Offset	Word	Fields
0x00	**REVISION** DI Information	**PRODREV** [15:8]: Production Revision **DEVINFOREV** [7:0]: DI Page Version
0x04	**EMBSIZE** Embedded Memory Information	**RAM** [27:16]: Internal RAM memory size **NVM** [15:0]: Internal NVM memory size
0x08	**STACKMSIZE** Co-Packaged Memory Information	**PSRAM** [31:16]: Co-packaged PSRAM Size **FLASH** [15:0]: Co-packaged Flash Size
0x0C	**SWCAPA** Software Restriction	**RFMCUEN** [31]: RF-MCU **NCPEN** [30]: NCP **GWEN** [29]: Gateway **ZWAVE** [12:10]: Z-Wave Capability • LEVEL0 (0): Z-Wave stack capability not available • LEVEL1 (1): Z-Wave and Z-Wave LR stack • LEVEL2 (2): Z-Wave stack capability without LR **RAIL** [9:8]: RAIL Capability • LEVEL0 (0): RAIL capability not available • LEVEL1 (1): RAIL enabled • LEVEL2 (2): N/A • LEVEL3 (3): N/A **CONNECT** [7:6]: Connect Capability • LEVEL0 (0): Connect stack capability not available • LEVEL1 (1): Connect enabled • LEVEL2 (2): N/A • LEVEL3 (3): N/A **BTSMART** [5:4]: Bluetooth Smart Capability • LEVEL0 (0): Bluetooth SMART stack capability not available • LEVEL1 (1): Bluetooth SMART enabled • LEVEL2 (2): N/A • LEVEL3 (3): N/A **THREAD** [3:2]: Thread Capability • LEVEL0 (0): Thread stack capability not available • LEVEL1 (1): Thread stack enabled • LEVEL2 (2): N/A • LEVEL3 (3): N/A **ZIGBEE** [1:0]: Zigbee Capability • LEVEL0 (0): Zigbee stack capability not available • LEVEL1 (1): GreenPower only • LEVEL2 (2): Zigbee and GreenPower • LEVEL3 (3): Zigbee only

Offset	Word	Fields
0x10	**HFRCOCALDEFAULT** Default HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value
0x14	**HFRCOCALSPEED** Fastest HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value
0x18	**EUI64L** EUI64 Low	**UNIQUEL** [31:0]: UNIQUEL
0x1C	**EUI64H** EUI64 High	**OUI** [31:8]: OUI **UNIQUEH** [7:0]: UNIQUEH
0x20	**PART0** Part Information	**DIECODE0** [31:24]: Die Code **SERIES** [23:16]: Series Identifier **TYPE** [15:8]: Device Type **PROTOCOL** [7:0]: Main Protocol • BLUETOOTH (66): Bluetooth • PROPRIETARY (70): Proprietary • FIFTEENPFOUR (77): 15.4 • PEARL (80): Pearl • WIFI (87): Wi-Fi • ZWAVE (90): Z-Wave
0x24	**PART1** Part Information	**FEATURE2** [31:24]: Device Feature • NOLED (48): LED option not available • LED (49): LED option available **FEATURE1** [23:16]: Device Feature **PACKAGE** [15:8]: Package Type **DIECODE1** [7:0]: Die Code
0x28	**PART2** External Component Info	**PINCOUNT** [31:24]: Pin Count **TEMPGRADE** [23:16]: Temperature Grade **FLASH** [15:8]: Flash Size **FEATURE3** [7:0]: Device Feature

Table 5.3. GP Fragment

Offset	Word	Fields
0x00	**TEMPERATURE** Calibration Temperature	**CALTEMP** [27:16]: Cal Temp **EMUTEMPROOM** [10:0]: Emu Room Temperature
0x04	**HFRCODPLLBAND0** HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value
0x08	**HFRCODPLLBAND1** HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value
0x0C	**HFRCODPLLBAND2** HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value
0x10	**HFRCODPLLBAND3** HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value

Offset	Word	Fields
0x14	**HFRCODPLLBAND4** HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value
0x18	**HFRCODPLLBAND5** HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value
0x1C	**HFRCODPLLBAND6** HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value
0x20	**HFRCODPLLBAND7** HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value

Offset	Word	Fields
0x24	**HFRCODPLLBAND8** HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value
0x28	**HFRCODPLLBAND9** HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value
0x2C	**HFRCODPLLBAND10** HFRCODPLL Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value
0x30	**HFRCOEM23DEFAULT** HFRCOEM23 Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value
0x34	**ADC0CALDATA** ADC0 Calibration Information	**TRIMGAIN0x3INT** [30:24]: Gain Calibration **TRIMGAIN4X** [22:16]: Gain Calibration **TRIMVROS1** [12:8]: Calibration Information **TRIMVROS0** [4:0]: Calibration Information

Offset	Word	Fields
0x38	**ADC0OFFSET** ADC0 Offset Information	**TRIMOFF4X** [21:16]: Offset trim **TRIMOFF2X** [13:8]: Offset trim **TRIMOFF1X** [5:0]: Offset trim
0x3C	**HFRCOEM2340MHZ** HFRCOEM23 Calibration Information	**TEMPCO** [31:28]: Tempco Trim **BIAS** [27:25]: Bias Current Setting **SIGMADELTA** [24:23]: Sigma Delta Setting **CLKDIV** [22:21]: HFRCO Clock Output Divider **BANDFINE** [20:15]: Band Fine Value **BANDCOARSE** [14:12]: Band Coarse Value **FINETUNING** [11:6]: Fine Tuning Value **TUNING** [5:0]: Coarse Tuning Value

5.6 SYSCFG - System Configuration

The SYSCFG block is a system-level register interface containing system status information, SYSTICK timer configuration, controls for several system exceptions, and four interrupt flags for general software use.

5.6.1 Software Interrupts

The SYSCFG contains four software interrupts (SW0 - SW3) that can be used to communicate between software tasks. Each software interrupt has a dedicated IRQ. The SWx control bits in SYSCFG_IEN are used to enable the individual software interrupts. To trigger a software interrupt set the corresponding bit in SYSCFG_IF.

5.6.2 FPU Exceptions

The SYSCFG block provides controls to enable various FPU exceptions in the SYSCFG_IEN register. Setting the FPIXC, FPIDC, FPOFC, FPUFC, FPDZC or FPIOC fields to '1' will enable the individual exception events. The SYSCFG_IF register contains the corresponding interrupt flag for each event.

5.6.3 Bus Faults

The SYSCFG_CTRL register provides controls to enable or disable specific bus fault events. ADDRFAULTEN enables bus faults on access to unmapped addresses in peripheral and NVM memory areas, and CLKDISFAULTEN enables bus faults on access to a peripheral whose bus clock is disabled.

5.6.4 SYSCFG Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x004	SYSCFG_IPVERSION	R	IP Version ID
0x008	SYSCFG_IF	RWH INTFLAG	Interrupt Flag
0x00C	SYSCFG_IEN	RW	Interrupt Enable
0x014	SYSCFG_CHIPREVHW	RWH	Chip Revision, Hard-Wired
0x018	SYSCFG_CHIPREV	RW	Chip Revision
0x024	SYSCFG_CFGSYSTIC	RW	SysTick Clock Source
0x034	SYSCFG_ROMREVHW	R	ROM Revision, Hard-Wire
0x200	SYSCFG_CTRL	RW	Control
0x600	SYSCFG_ROOTDATA0	RW	Data Register 0
0x604	SYSCFG_ROOTDATA1	RW	Data Register 1
0x608	SYSCFG_ROOTLOCKSTATUS	RH	Lock Status
0x60C	SYSCFG_ROOTSESWVERSION	RW	SE SW Version
0x610	SYSCFG_ROOTSEROOTINFO	RW	SE Root Info
0x614	SYSCFG_ROOTSEROOTINFOSTATUS	R	SE Root Info Status
0x1004	SYSCFG_IPVERSION_SET	R	IP Version ID
0x1008	SYSCFG_IF_SET	RWH INTFLAG	Interrupt Flag
0x100C	SYSCFG_IEN_SET	RW	Interrupt Enable
0x1014	SYSCFG_CHIPREVHW_SET	RWH	Chip Revision, Hard-Wired
0x1018	SYSCFG_CHIPREV_SET	RW	Chip Revision
0x1024	SYSCFG_CFGSYSTIC_SET	RW	SysTick Clock Source
0x1034	SYSCFG_ROMREVHW_SET	R	ROM Revision, Hard-Wire
0x1200	SYSCFG_CTRL_SET	RW	Control
0x1600	SYSCFG_ROOTDATA0_SET	RW	Data Register 0
0x1604	SYSCFG_ROOTDATA1_SET	RW	Data Register 1
0x1608	SYSCFG_ROOTLOCKSTATUS_SET	RH	Lock Status
0x160C	SYSCFG_ROOTSESWVERSION_SET	RW	SE SW Version
0x1610	SYSCFG_ROOTSEROOTINFO_SET	RW	SE Root Info
0x1614	SYSCFG_ROOTSEROOTINFOSTATUS_SET	R	SE Root Info Status
0x2004	SYSCFG_IPVERSION_CLR	R	IP Version ID
0x2008	SYSCFG_IF_CLR	RWH INTFLAG	Interrupt Flag
0x200C	SYSCFG_IEN_CLR	RW	Interrupt Enable
0x2014	SYSCFG_CHIPREVHW_CLR	RWH	Chip Revision, Hard-Wired
0x2018	SYSCFG_CHIPREV_CLR	RW	Chip Revision
0x2024	SYSCFG_CFGSYSTIC_CLR	RW	SysTick Clock Source
0x2034	SYSCFG_ROMREVHW_CLR	R	ROM Revision, Hard-Wire

Offset	Name	Type	Description
0x2200	SYSCFG_CTRL_CLR	RW	Control
0x2600	SYSCFG_ROOTDATA0_CLR	RW	Data Register 0
0x2604	SYSCFG_ROOTDATA1_CLR	RW	Data Register 1
0x2608	SYSCFG_ROOTLOCKSTA-TUS_CLR	RH	Lock Status
0x260C	SYSCFG_ROOTSESWVER-SION_CLR	RW	SE SW Version
0x2610	SYSCFG_ROOTSEROOTIN-FO_CLR	RW	SE Root Info
0x2614	SYSCFG_ROOTSEROOTIN-FOSTATUS_CLR	R	SE Root Info Status
0x3004	SYSCFG_IPVERSION_TGL	R	IP Version ID
0x3008	SYSCFG_IF_TGL	RWH INTFLAG	Interrupt Flag
0x300C	SYSCFG_IEN_TGL	RW	Interrupt Enable
0x3014	SYSCFG_CHIPREVHW_TGL	RWH	Chip Revision, Hard-Wired
0x3018	SYSCFG_CHIPREV_TGL	RW	Chip Revision
0x3024	SYSCFG_CFGSYSTIC_TGL	RW	SysTick Clock Source
0x3034	SYSCFG_ROMREVHW_TGL	R	ROM Revision, Hard-Wire
0x3200	SYSCFG_CTRL_TGL	RW	Control
0x3600	SYSCFG_ROOTDATA0_TGL	RW	Data Register 0
0x3604	SYSCFG_ROOTDATA1_TGL	RW	Data Register 1
0x3608	SYSCFG_ROOTLOCKSTA-TUS_TGL	RH	Lock Status
0x360C	SYSCFG_ROOTSESWVER-SION_TGL	RW	SE SW Version
0x3610	SYSCFG_ROOTSEROOTIN-FO_TGL	RW	SE Root Info
0x3614	SYSCFG_ROOTSEROOTIN-FOSTATUS_TGL	R	SE Root Info Status

5.6.5 SYSCFG Register Description

5.6.5.1 SYSCFG_IPVERSION - IP Version ID

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xA																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0xA	R	**IP Version**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

5.6.5.2 SYSCFG_IF - Interrupt Flag

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	0x0			0x0	0x0	0x0	0x0	0x0	0x0					0x0	0x0	0x0	0x0
Access															RW	RW			RW	RW	RW	RW	RW	RW					RW	RW	RW	RW
Name															L1CACHE0	TEALSYS			FPIXC	FPIDC	FPOFC	FPUFC	FPDZC	FPIOC					SW3	SW2	SW1	SW0

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	L1CACHE0	0x0	RW	**L1CACHE bus error**
	L1CACHE bus error			
16	TEALSYS	0x0	RW	**TEALSYS bus error**
	TEALSYS Host fabric bus error			
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	FPIXC	0x0	RW	**FPU Inexact interrupt flag**
	Set upon FPU inexact exception			
12	FPIDC	0x0	RW	**FPU Input denormal interrupt flag**
	Set upon FPU input denormal exception			
11	FPOFC	0x0	RW	**FPU Overflow interrupt flag**
	Set upon FPU overflow exception			
10	FPUFC	0x0	RW	**FPU Underflow interrupt flag**
	Set upon FPU underflow exception			
9	FPDZC	0x0	RW	**FPU Divide by zero interrupt flag**
	Set upon FPU divide by zero operation			
8	FPIOC	0x0	RW	**FPU Invalid Operation interrupt flag**
	Set upon FPU invalid operation			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	SW3	0x0	RW	**Software Interrupt Flag**
	Software interrupts			
2	SW2	0x0	RW	**Software Interrupt Flag**
	Software interrupts			
1	SW1	0x0	RW	**Software Interrupt Flag**
	Software interrupts			
0	SW0	0x0	RW	**Software Interrupt Flag**
	Software interrupts			

5.6.5.3 SYSCFG_IEN - Interrupt Enable

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	0x0			0x0	0x0	0x0	0x0	0x0	0x0					0x0	0x0	0x0	0x0
Access															RW	RW			RW	RW	RW	RW	RW	RW					RW	RW	RW	RW
Name															L1CACHE0	TEALSYS			FPIXC	FPIDC	FPOFC	FPUFC	FPDZC	FPIOC					SW3	SW2	SW1	SW0

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	L1CACHE0	0x0	RW	**L1CACHE bus error enable**
	L1CACHE bus error interrupt enable			
16	TEALSYS	0x0	RW	**TEALSYS bus error enable**
	TEALSYS Host fabric bus error interrupt enable			
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	FPIXC	0x0	RW	**FPU Inexact Interrupt Enable**
	Set to enable the FPIXCIF Interrupt			
12	FPIDC	0x0	RW	**FPU Input denormal Interrupt Enable**
	Set to enable the FPIDCIF Interrupt			
11	FPOFC	0x0	RW	**FPU Overflow Interrupt Enable**
	Set to enable the FPOFCIF Interrupt			
10	FPUFC	0x0	RW	**FPU Underflow Interrupt Enable**
	Set to enable the FPUFCIF Interrupt			
9	FPDZC	0x0	RW	**FPU Divide by zero Interrupt Enable**
	Set to enable the FPDZCIF Interrupt			
8	FPIOC	0x0	RW	**FPU Invalid Operation Interrupt Enable**
	Set to enable the FPIOCIF Interrupt			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	SW3	0x0	RW	**Software Interrupt Enable**
	Set to enable the Software Interrupts			
2	SW2	0x0	RW	**Software Interrupt Enable**
	Set to enable the Software Interrupts			
1	SW1	0x0	RW	**Software Interrupt Enable**
	Set to enable the Software Interrupts			
0	SW0	0x0	RW	**Software Interrupt Enable**
	Set to enable the Software Interrupts			

5.6.5.4 SYSCFG_CHIPREVHW - Chip Revision, Hard-Wired

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x2				0x0				0x30											
Access													R				R				RW											
Name													MAJOR				MINOR				PARTNUMBER											

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	MAJOR	0x2	R	**Hardwired Chip Major value** Stores the Hardwired Chip Revision Minor signal value upper 4bits, and reflects tie-block value that can change from revision to revision.
15:12	MINOR	0x0	R	**Hardwired Chip Minor value** Stores the Hardwired Chip Revision Minor signal value lower 4 bits.
11:0	PARTNUMBER	0x30	RW	**Chip Part Number value**

5.6.5.5 SYSCFG_CHIPREV - Chip Revision

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0				0x0				0x0											
Access													RW				RW				RW											
Name													MAJOR				MINOR				PARTNUMBER											

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	MAJOR	0x0	RW	**Chip Revision Major value** Upper 4bits of Chip Revision Minor
15:12	MINOR	0x0	RW	**Chip Revision Minor value** Lower 4bits of Chip Revision Minor
11:0	PARTNUMBER	0x0	RW	**Chip Part Number value**

5.6.5.6 SYSCFG_CFGSYSTIC - SysTick Clock Source

Offset	Bit Position																															
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																SYSTICEXTCLKEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	SYSTICEXTCLKEN	0x0	RW	**SysTick External Clock Enable** Set 1 to use an external clock as the M33 System Tick.

5.6.5.7 SYSCFG_ROMREVHW - ROM Revision, Hard-Wire

Offset	Bit Position																															
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x30																								0x5							
Access	R																								R							
Name	SECOMPATID																								SEROMREV							

Bit	Name	Reset	Access	Description
31:24	SECOMPATID	0x30	R	**SECOMPATID** Store the Hardwired SE firmware die compatibilty ID
23:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SEROMREV	0x5	R	**SEROMREV** Store the Hard-wired ROM revision.

5.6.5.8 SYSCFG_CTRL - Control

Offset	Bit Position																													
0x200	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	0x1
Access																															RW	RW
Name																															CLKDISFAULTEN	ADDRFAULTEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CLKDISFAULTEN	0x1	RW	**Disabled Clkbus Bus Fault Enable**
	When this bit is set, busfaults are generated on accesses to peripherals with disabled bus clock			
0	ADDRFAULTEN	0x1	RW	**Invalid Address Bus Fault Response Enabl**
	When this bit is set, busfaults are generated on accesses to unmapped parts of system and code address space			

5.6.5.9 SYSCFG_ROOTDATA0 - Data Register 0

Offset	Bit Position																													
0x600	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	RW	**Data**
	Generic data space for user to pass to root, e.g., address of struct in mem			

5.6.5.10 SYSCFG_ROOTDATA1 - Data Register 1

Offset	Bit Position																													
0x604	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	RW	**Data**
	Generic data space for user to pass to root, e.g., address of struct in mem			

5.6.5.11 SYSCFG_ROOTLOCKSTATUS - Lock Status

Offset	Bit Position																														
0x608	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0									0x1	0x1	0x1	0x1	0x1	0x1	0x1								0x1						0x1	0x1	0x1
Access	R									R	R	R	R	R	R	R								R						R	R	R
Name	EFUSEUNLOCKED									LPWCPU1DBGLOCK	LPWCPU0DBGLOCK	USERSPNIDLOCK	USERSPIDLOCK	USERNIDLOCK	USERDBGLOCK	USERDBGAPLOCK								ROOTDBGLOCK						MFRLOCK	REGLOCK	BUSLOCK

Bit	Name	Reset	Access	Description
31	EFUSEUNLOCKED	0x0	R	**E-Fuse Unlocked**
	E-Fuse Unlocked when 1; Locked when 0.			
30:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	LPWCPU1DBGLOCK	0x1	R	**LPW CPU1 Debug Lock**
	Locked when 1; unlocked when 0.			
21	LPWCPU0DBGLOCK	0x1	R	**LPW CPU0 Debug Lock**
	Locked when 1; unlocked when 0.			
20	USERSPNIDLOCK	0x1	R	**User Secure Non-invasive Debug Lock**
	Locked when 1; unlocked when 0.			
19	USERSPIDLOCK	0x1	R	**User Secure Invasive Debug Lock**
	Locked when 1; unlocked when 0.			
18	USERNIDLOCK	0x1	R	**User Non-invasive Debug Lock**
	Locked when 1; unlocked when 0.			
17	USERDBGLOCK	0x1	R	**User Invasive Debug Lock**
	Locked when 1; unlocked when 0.			
16	USERDBGAPLOCK	0x1	R	**User Debug Access Port Lock**
	Locked when 1; unlocked when 0.			
15:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	ROOTDBGLOCK	0x1	R	**Root Debug Lock**
	Locked when 1; unlocked when 0.			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	MFRLOCK	0x1	R	**Manufacture Lock**
	Locked when 1; unlocked when 0.			
1	REGLOCK	0x1	R	**Register Lock**
	Locked when 1; unlocked when 0.			
0	BUSLOCK	0x1	R	**Bus Lock**
	Locked when 1; unlocked when 0.			

5.6.5.12 SYSCFG_ROOTSESWVERSION - SE SW Version

Offset	Bit Position																													
0x60C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	SWVERSION																															

Bit	Name	Reset	Access	Description
31:0	SWVERSION	0x0	RW	**SW Version**
	Location for SE to write the Firmware version and host to read it			

5.6.5.13 SYSCFG_ROOTSEROOTINFO - SE Root Info

Offset	Bit Position																													
0x610	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																						0x0	0x0	0x0								0x0
Access																						RW	RW	RW								RW
Name																						ROOTSELIFECYCLE2	ROOTSELIFECYCLE1	ROOTSELIFECYCLE0								ROOTSECOMPROMISED

Bit	Name	Reset	Access	Description
31:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	ROOTSELIFECYCLE2	0x0	RW	**ROOTSELIFECYCLE2**
	Write-once root only register.			
9	ROOTSELIFECYCLE1	0x0	RW	**ROOTSELIFECYCLE1**
	Write-once root only register.			
8	ROOTSELIFECYCLE0	0x0	RW	**ROOTSELIFECYCLE0**
	Write-once root only register.			
7:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	ROOTSECOMPRO-MISED	0x0	RW	**ROOT SE COMPROMISED**
	Write-once root only register.			

5.6.5.14 SYSCFG_ROOTSEROOTINFOSTATUS - SE Root Info Status

Offset	Bit Position																													
0x614	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																						0x0	0x0	0x0								0x0
Access																						R	R	R								R
Name																						SELIFECYCLE2	SELIFECYCLE1	SELIFECYCLE0								SECOMPROMISED

Bit	Name	Reset	Access	Description
31:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	SELIFECYCLE2	0x0	R	**SELIFECYCLE2**
	SELIFECYCLE2 status value			
9	SELIFECYCLE1	0x0	R	**SELIFECYCLE1**
	SELIFECYCLE1 status value			
8	SELIFECYCLE0	0x0	R	**SELIFECYCLE0**
	SELIFECYCLE0 status value			
7:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	SECOMPROMISED	0x0	R	**SECOMPROMISED**
	SECOMPROMISED status value			

5.6.6 SYSCFG_CFGNS Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x01C	SYSCFG_CFGNSTCALIB	RW	Configure Non-Secure Sys-Tick Cal.
0x600	SYSCFG_ROOTNSDATA0	RW	Data Register 0
0x604	SYSCFG_ROOTNSDATA1	RW	Data Register 1
0x101C	SYSCFG_CFGNSTCALIB_SET	RW	Configure Non-Secure Sys-Tick Cal.
0x1600	SYSCFG_ROOTNSDATA0_SET	RW	Data Register 0
0x1604	SYSCFG_ROOTNSDATA1_SET	RW	Data Register 1
0x201C	SYSCFG_CFGNSTCALIB_CLR	RW	Configure Non-Secure Sys-Tick Cal.
0x2600	SYSCFG_ROOTNSDATA0_CLR	RW	Data Register 0
0x2604	SYSCFG_ROOTNSDATA1_CLR	RW	Data Register 1
0x301C	SYSCFG_CFGNSTCALIB_TGL	RW	Configure Non-Secure Sys-Tick Cal.
0x3600	SYSCFG_ROOTNSDATA0_TGL	RW	Data Register 0
0x3604	SYSCFG_ROOTNSDATA1_TGL	RW	Data Register 1

5.6.7 SYSCFG_CFGNS Register Description

5.6.7.1 SYSCFG_CFGNSTCALIB - Configure Non-Secure Sys-Tick Cal.

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset							0x0	0x1	0x0																					
Access							RW	RW	RW																					
Name							NOREF	SKEW	TENMS																					

Bit	Name	Reset	Access	Description
31:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	NOREF	0x0	RW	**No Reference**
	Indicates whether external reference clock is implemented.			
	Value	Mode		Description
	0	REF		Reference clock is implemented
	1	NOREF		Reference clock is not implemented
24	SKEW	0x1	RW	**Skew**
	Set this bit to 1, if the external reference clock can not guarantee an exact multiple of 10ms.			
23:0	TENMS	0x0	RW	**Ten Milliseconds**
	Provide an integer value to compute a 10ms delay from either the external reference clock or system clock. By default at reset with system clock at 19MHz, with 0x4a37, 1ms is generated.			

5.6.7.2 SYSCFG_ROOTNSDATA0 - Data Register 0

Offset	Bit Position																													
0x600	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																													
Access	RW																													
Name	DATA																													

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	RW	**Data**
	Generic data space for user to pass to root, e.g., address of struct in mem			

5.6.7.3 SYSCFG_ROOTNSDATA1 - Data Register 1

Offset	Bit Position																													
0x604	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	RW	**Data**
	Generic data space for user to pass to root, e.g., address of struct in mem			

5.7 SEMAPHORE

5.7.1 Introduction

There are often variables or data structures that different processor threads or different cores need to operate on simultaneously. The SEMPAHORE block provides a simple register interface that software can use to eliminate contention between different parts of the system accessing data at the same time.

5.7.2 Description

Each SEMAPHORE block provides 128 external self-locking registers to enable resource sharing. The LOCKx registers are not tied to specific memory regions or peripherals. An example LOCK register is shown in 5.7.2.1 SEMAPHORE_LOCKx - LOCKx Register. Lock registers are numbered as LOCK0 - LOCK127 in the register map.

The usage of the LOCKx registers is simple. A read operation will lock the register, and a write operation will unlock the register.

When an unlocked register is read, the read will return a '0', and SEMAPHORE will immediately set the lock on that register. Any subsequent reads while the register is locked will return a '1'. When the register is written, the lock will be released.

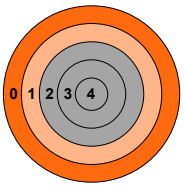
Software can use this behavior to prevent contention on any data structure. Any data structure to be protected should be assigned a shared LOCKx register location that all accessing functions will use. Before modifying any shared data structure, software performs a read of the corresponding LOCKx register. If the read returns '0', software can consider the structure locked and available to modify as needed. When the required access is complete, software must write the LOCKx register to reset the lock and allow other threads access. If the LOCKx register read returns '1' it means another thread is busy accessing the assigned data structure, and the current thread must try again at a later time.

5.7.2.1 SEMAPHORE_LOCKx - LOCKx Register

Offset	Bit Position																														
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																LOCKx

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	LOCKx	0x0	RW	**SEMAPHORE LOCK x Bit Register** This bit gives the status of the lock when read. '0'- lock acquired '1'- lock acquire attempt unsuccessful. To release the lock this bit is written (the write value is ignored).

6. Low-Power Wireless Transceiver (LPW)



Quick Facts

What?

The LPW subsystem is a radio transceiver providing access to transmit and receive data, radio settings and control interface.

Why?

The LPW enables the user to communicate using a wide range of data rates, modulation and frame formats.

How?

Dynamic or fixed frame lengths, optional address recognition, flexible CRC and crypto schemes makes the SiXG301 perfectly suit any application using low or medium data rate radio communication.

6.1 Introduction

The Low-Power Wireless (LPW) subsystem of the SiXG301 enables the user to control a wide range of settings and options for tailoring radio operation precisely to the users need. It provides access to the transmit and receive data buffers and supports both dynamic and static frame lengths, as well as automatic address filtering and CRC insertion/verification.

As seen in Figure 6.1. LPW Radio Subsystem Overview, the LPW subsystem consists of a radio processor core, RAM, and several modules all responsible for specific tasks. Please refer to the abbreviations section (Appendix A Abbreviations) for a description of acronyms.

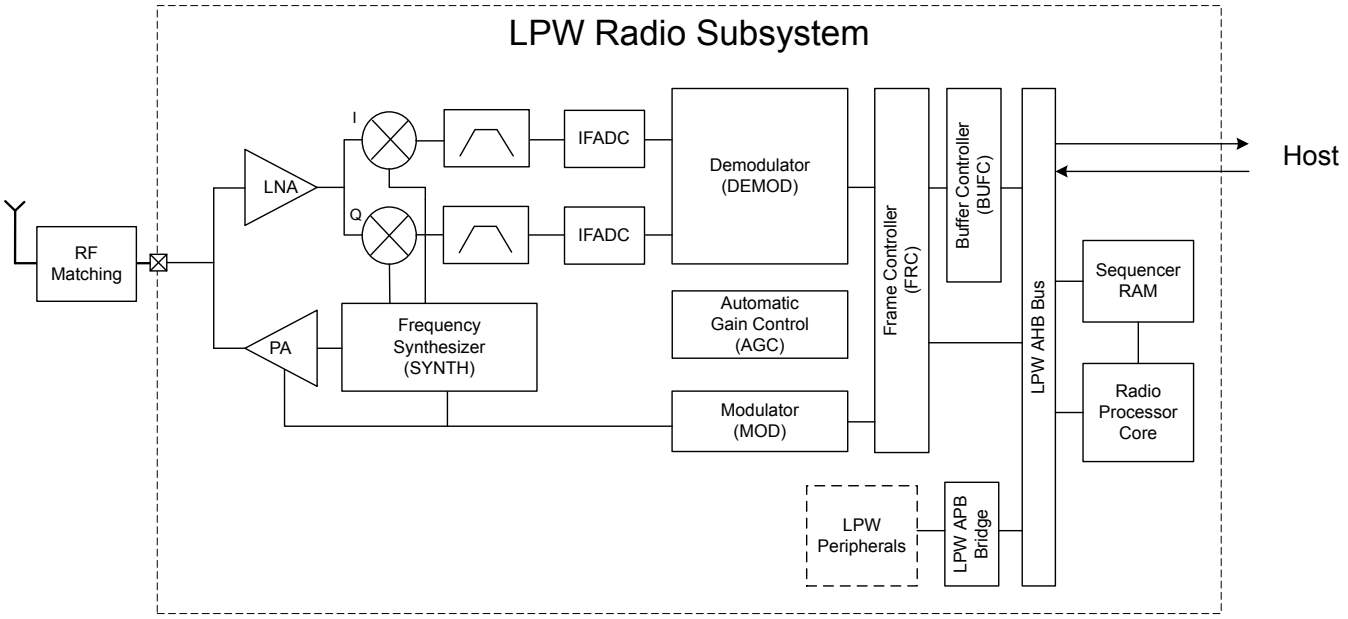


Figure 6.1. LPW Radio Subsystem Overview

During transmission (TX), the Radio Controller enables the Frequency Synthesizer (SYNTH), Modulator (MOD), and Power Amplifier (PA). The Modulator requests data from the Frame Controller (FRC), which reads data from a buffer. Based upon modulation format and data to send, the Modulator manipulates the SYNTH to output the correct frequency and phase. When the whole frame has been transmitted, the radio can automatically switch to receive mode.

In receive mode (RX), the radio controller enables the Low Noise Amplifier (LNA), Frequency Synthesizer (SYNTH), Mixers, Intermediate Frequency ADCs (IFADC), and Demodulator (DEMOD). The Demodulator searches for valid frames according to modulation format and data rate. If a frame is detected, the demodulated data is handed to the Frame Controller (FRC), which stores the data in the Buffer. When the complete frame has been received (determined by the Frame Controller), it is possible to either go to TX or stay in RX to search for a new frame.

The Radio Transceiver interface is accessible through software drivers provided by Silicon Labs.

6.1.1 RF Frequency Synthesizer

The Fractional-N RF Frequency Synthesizer (SYNTH) provides a low phase noise LO signal to be used in both receive and transmit modes.

The capabilities of the SYNTH include:

- High performance, low phase noise
- Fast frequency settling
- Fast and fully automated calibration
- Sub 100 Hz frequency resolution across the supported frequency bands

6.1.2 Modulation Modes

SiXG301 supports a wide range of modulation modes in transmit and receive:

- 2-FSK, 2-GFSK, 4-FSK, MSK, GMSK, O-QPSK with half-sine shaping, ASK / OOK, DBPSK TX
- NRZ or Manchester support
- UART mode over air for legacy protocols
- Baudrates ranging from below 100 Baud/s to 2 MBaud/s, allowing data rates up to 4 MBit/s
- Configurable frequency deviation
- Configurable Direct Sequence Spread Spectrum (DSSS), with spread sequences up to 32 chips encoding up to 4 information bits
- Configurable 4-FSK symbol encoding

6.1.3 Transmit Mode

In transmit mode SiXG301 performs the following functionality:

- Automatic PA power ramping during the start and end of a frame transmit
- Programmable output power
- Optional preamble and synchronization word insertion
- Accurate transmit frame timing to support time synchronized radio protocols
- Optional Carrier Sense Multiple Access - Collision Avoidance (CSMA-CA) or Listen Before Talk (LBT) hardware support
- Integrated transmit test modes, as described in 6.1.12 RF Test Modes

6.1.4 Receive Mode

In receive mode SiXG301 performs the following functionality:

- A single-ended LNA amplifies the input RF signal. The amplified signal is then mixed to a low-IF signal through the quadrature down-coversion mixer. Further signal filtering is performed before conversion to a digital signal through the I/Q ADC.
- Digitally configurable receiver bandwidth from 100 Hz to 2.5 MHz
- Timing recovery on received data, including simultaneous support for two different frame synchronization words
- Automatic frequency offset compensation, to compensate for carrier frequency offset between the transmitter and receiver
- Support for a wide range of modulation formats as described in section 6.1.2 Modulation Modes

6.1.5 Data Buffering

SiXG301 supports buffered transmit and receive modes through its buffer controller (BUFC), with four individually configurable buffers. The BUFC uses the system RAM as storage, and each buffer can be individually configured with parameters such as:

- Buffer size
- Buffer interrupt thresholds
- Buffer RAM location
- Overflow and underflow detection

In receive mode, data following frame synchronization is moved directly from the demodulator to the buffer storage.

In transmit mode, data following the inserted preamble and synchronization word is moved directly from the buffer storage to the modulator.

6.1.6 Unbuffered Data Transfer

For most system designs it is recommended to use the data buffering within SiXG301 to provide a convenient user interface.

In unbuffered data transfer modes the hardware support provided by SiXG301 to perform preamble and frame synchronization insertion in transmit mode and detection in receive mode can still optionally be used.

6.1.7 Frame Format Support

SiXG301 has an extensive support for frame handling in transmit and receive modes, which allows effective handling of even advanced protocols. The frame format support is controlled by the Frame Controller (FRC). The support includes:

- Preamble and frame synchronization inserted into transmitted frames
- Full frame synchronization of received frames
- Simple address matching of received frames in hardware, further configurable address and frame filtering supported through sequencer
- Support for variable length frames
- Automated CRC calculation and verification
- Configurable bit ordering, with the most or least significant bit transmitted and received first

6.1.8 Hardware CRC Support

SiXG301 supports a configurable CRC generation in transmit and verification in receive mode:

- 8, 16, 24 or 32 bit CRC value
- Configurable polynomial and initialization value
- Optional inversion of CRC value over air
- Configurable CRC byte ordering
- Support for multiple CRC values calculated and verified per transmitted or received frame
- The CRC module is typically controlled by the Frame Controller (FRC) for in-line operations in transmit and receive modes. Alternatively, the CRC module may be accessed directly from software to calculate and verify CRC data.

6.1.9 Convolutional Encoding / Decoding

SiXG301 includes hardware support for convolutional encoding and decoding, for forward error correction (FEC). This feature is performed by the Frame Controller (FRC) module:

- Constraint length configurable up to 7, for the highest robustness
- Configurable puncturing, to achieve rates between 1/2 rate and full rate
- Configurable soft decision or hard decision decoding
- Convolutional coding may be used together with the symbol interleaver to improve robustness against burst errors

6.1.10 Binary Block Encoding / Decoding

SiXG301 includes hardware support for binary block encoding and decoding, both performed real-time in the the transmit and receive path. This is performed in the Frame Controller (FRC) module:

The block coding works on blocks of up to 16 bits of data and adds parity bits to be capable of single or multiple bit corrections by the receiver.

- One or more parity bits can be added and verified
- Bit error correction
- Lookup-codes can be used to implement virtually any block coding scheme

6.1.11 Data Encryption and Authentication

SiXG301 has hardware support for AES encryption, decryption and authentication modes. These security operations can be performed on data in RAM or any data buffer, without further CPU intervention. The key size is 128 bits.

AES modes of operations directly supported by the SiXG301 hardware are listed in Table 6.1. AES modes of operation with hardware support. In addition to these modes, other modes can also be implemented by using combinations of modes. For example, the CCM mode can be implemented using the CTR and CBC-MAC modes in combination.

Table 6.1. AES modes of operation with hardware support

AES Mode	Encryption / Decryption	Authentication	Comment
ECB	Yes	—	Electronic Code Book
CTR	Yes	—	Counter mode
CCM	Yes	Yes	Counter with CBC-MAC
CCM*	Yes	Yes	CCM with encryption-only and integrity-only capabilities
GCM	Yes	Yes	Galois Counter Mode
CBC	Yes	—	Cipher Block Chaining
CBC-MAC	—	Yes	Cipher Block Chaining, Message Authentication Code
CMAC	—	Yes	Cipher-based MAC
CFB	Yes	—	Cipher Feedback
OFB	Yes	—	Output Feedback

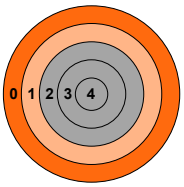
The Cryptographic Acceleration module can operate directly on data buffers provided by the buffer controller (BUFC) module. It is also possible to provide data directly from the embedded Cortex-M33 or via DMA.

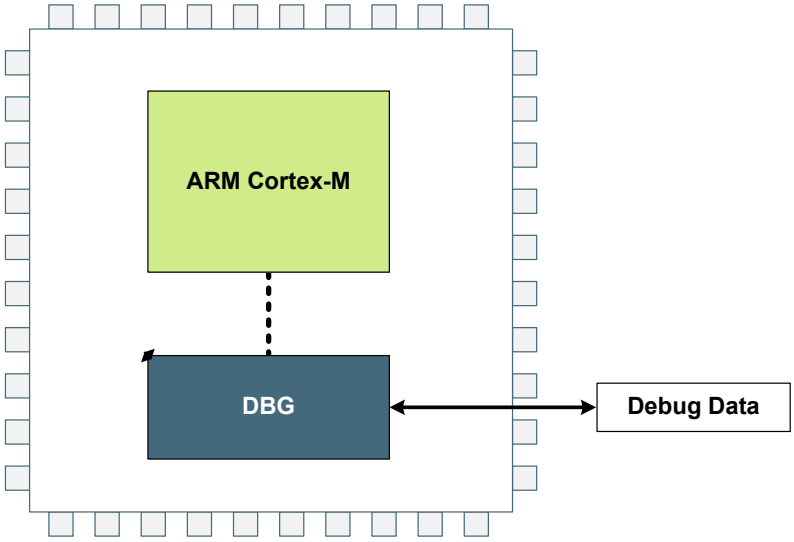
6.1.12 RF Test Modes

SiXG301 supports a wide range of RF test modes typically used for characterization and regulation compliance testing, including:

- Unmodulated carrier transmit
- Modulated carrier transmit, with internal configurable pseudo random data generator
- Continuous data reception for Bit Error Rate (BER) measurements
- Storing of raw receiver data to RAM
- Transmit of raw frequency data from RAM

7. DBG - Debug Interface





Quick Facts

What?

The DBG is used to program and debug SiXG301 devices.

Why?

The DBG makes it easy to re-program and update the system in the field, and allows debugging with minimal I/O pin usage.

How?

The Cortex-M33 supports advanced debugging features. SiXG301 devices can use a minimum of two port pins for debugging or programming. The internal and external state of the system can be examined with debug extensions supporting instruction or data access break and watch points.

7.1 Introduction

The SiXG301 devices include hardware debug support through a 2-pin serial-wire debug (SWD) interface or a 4-pin Joint Test Action Group (JTAG) interface, as well as an Embedded Trace Module (ETM) for data/instruction tracing. In addition, there is also a Serial Wire Viewer pin which can be used to output profiling information, data trace and software-generated messages.

For more technical information about the debug interface the reader is referred to:

- ARM Cortex-M33 Technical Reference Manual
- ARM CoreSight Components Technical Reference Manual
- ARM Debug Interface v5 Architecture Specification
- IEEE Standard for Test Access Port and Boundary-Scan Architecture, IEEE 1149.1-2013

7.2 Features

- Debug Access Port Serial Wire JTAG (DAPSWJ)
  - Implements the ADIv5 debug interface
- ARM Trustzone
  - Enables secure debugging
- Breakpoint unit (BPU)
  - Implement up to 8 hardware breakpoints
- Data Watch point and Trace (DWT) unit
  - Implement up to 4 watch points, trigger resources and system profiling
- Instrumentation Trace Macrocell (ITM)
  - Application-driven trace source that supports printf style debugging
- Embedded Trace Macrocell v4.2 (ETM)
  - Real time instruction and data trace information of the processor
- Cross Trigger Interface (CTI)
  - Issues synchronous triggers based on system events
  - Can be used to generate IRQs or route to PRS signalling

7.3 Functional Description

There are debug and trace pins available on the device. Operation of these pins is described in the following sections.

7.3.1 Debug Pins

The following pins are the debug connections for the device:

- Serial Wire Clock Input (SWCLK) : This pin is enabled after power-up and has a built-in pull-down.
- Serial Wire Data Input/Output (SWDIO) : This pin is enabled after power-up and has a built-in pull-up.
- Test Data Output (TDO): This pin is assigned to JTAG functionality after power-up. However, it remains in high-Z state until the first valid JTAG command is received.
- Test Data Input (TDI): This pin is assigned to JTAG functionality after power-up. However, it remains in high-Z state until the first valid JTAG command is received. Once enabled, the pin has a built-in pull-up.
- Serial Wire Viewer (SWV): This pin is disabled after reset.

The debug pins have integrated pull devices that are enabled by default after a reset. Leaving them enabled may increase current consumption if the pins are connected to power or ground. The debug pins have enable bits in the GPIO_DBGROUTEPEN register; refer to the GPIO chapter for more details. Upon disabling the debug pins, debug contact with the device is lost once the DAPSWJ power request bits are deasserted. By default after a power cycle, the DAPSWJ is in JTAG mode. If during a debugging session the device is switched to SWD mode, a power cycle is needed to return to JTAG mode.

7.3.2 Embedded Trace Macrocell (ETM)

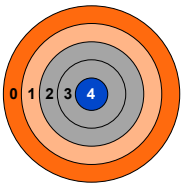
ETM makes it possible to non-intrusively trace both instruction and data from the processor in real time. Trace can be controlled through a set of triggering and filtering resources. The resources include 4 address comparators, 2 data value comparators, 2 counters, a context ID comparator and a sequencer. Before enabling the ETM, the CMU_TRACECLKCTRL register must be configured to select the desired trace clock source. See 8. CMU - Clock Management Unit for details.

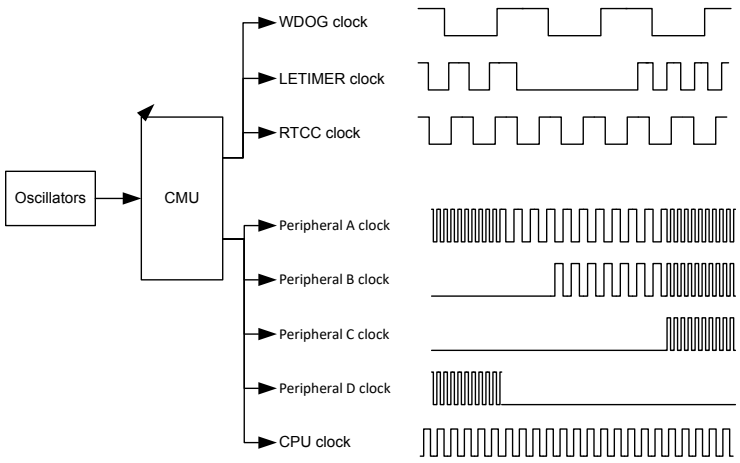
The trace can be exported through a set of trace pins, which include:

- Trace Clock (GPIO.TRACECLK): Functions as a sample clock for the trace. This pin is disabled after reset.
- Trace Data 0-3 (GPIO.TRACEDATA0, GPIO.TRACEDATA1, GPIO.TRACEDATA2, GPIO.TRACEDATA3): The trace data pins provide the compressed trace stream. These pins are disabled after reset.

For information on how to configure the ETM, see the ARM Embedded Trace Macrocell Architecture Specification. The Trace Clock and Trace Data pins are enabled through a GPIO register. For more information on how to enable the ETM pins, refer to the GPIO chapter.

8. CMU - Clock Management Unit





Quick Facts

What?

The CMU controls clock switching and distribution for the host processor core and peripherals. SiXG301 supports several different oscillators with minimized power consumption and short start-up time. The CMU has hardware support for calibration of the tuneable RC oscillators.

Why?

Oscillators and clocks contribute significantly to the power consumption of the MCU. Low power oscillators combined with a flexible clock control scheme help to minimize the energy consumption in a given application.

How?

The CMU switches different clock sources for various peripherals, and sets the clock for the core and bus clocks. The short oscillator start-up time makes duty-cycling between active mode and the low energy modes very efficient. Independent clock trees for different peripheral sets ensure that critical timing can be met, while scaling core and bus clocks to save energy.

8.1 Introduction

The Clock Management Unit (CMU) is responsible for switching among various oscillator sources and provides clocks to the peripheral modules. Oscillators and clock trees are automatically turned on and off based on demand from the peripherals to minimize power consumption.

8.2 Features

- Multiple clock sources available:
  - 150 MHz High Frequency SoC PLL (SOCPLL)
  - 38 MHz - 40 MHz High Frequency Crystal Oscillator (HFXO)
  - 1 MHz - 100 MHz High Frequency RC Oscillator (HFRCO0 / HFRCODPLL)
  - 1 MHz - 40 MHz High Frequency RC Oscillator (HFRCOEM23)
  - 20 MHz Fast Startup RC Oscillator (FSRCO)
  - 2 MHz - 38 MHz External Clock from Input Pin (CLKIN0)
  - 32.768 kHz Low Frequency Crystal Oscillator (LFXO)
  - 32.768 kHz Low Frequency RC Oscillator (LFRCO) with Precision Mode
  - 1000 Hz Ultra Low Frequency RC Oscillator (ULFRCO)
- On-demand oscillator request.
- Low power oscillators.
- Fast start-up times.
- Cascaded prescalers from SYSCLK to AHB Clock (HCLK), APB Clocks (PCLK, LSPCLK), and System mailbox clock (SYSMBCLK)
- Independent, asynchronous clock trees for blocks of peripehrals.
- Clock gating on an individual basis to all peripherals based on module enable.
- Selectable clocks can be output on external pins and/or PRS.
- Hardware support for calibration of RC oscillators.

8.3 Functional Description

The CMU is comprised of several programmable clock trees, which connect oscillator resources to peripherals and buses. This section describes clock sources and peripherals available to the largest devices in the SiXG301 family. Please refer to the Configuration Summary in the Device Data Sheet to see which core and peripheral modules, and therefore clock connections, are present in a specific device. Bus clock selection, including peripherals clocked directly from bus clocks, is shown in Figure 8.1. Bus Clocks. Clock selection for peripherals with multiple high-frequency clock sources is shown in Figure 8.2. High-Frequency Peripheral Clocks. Clock selection for peripherals with multiple low-frequency clock sources is shown in Figure 8.3. Low-Frequency Peripheral Clocks. Clock selection for peripherals that can select from a high or low frequency clock source is shown in Figure 8.4. Mixed-Frequency Peripheral Clocks.

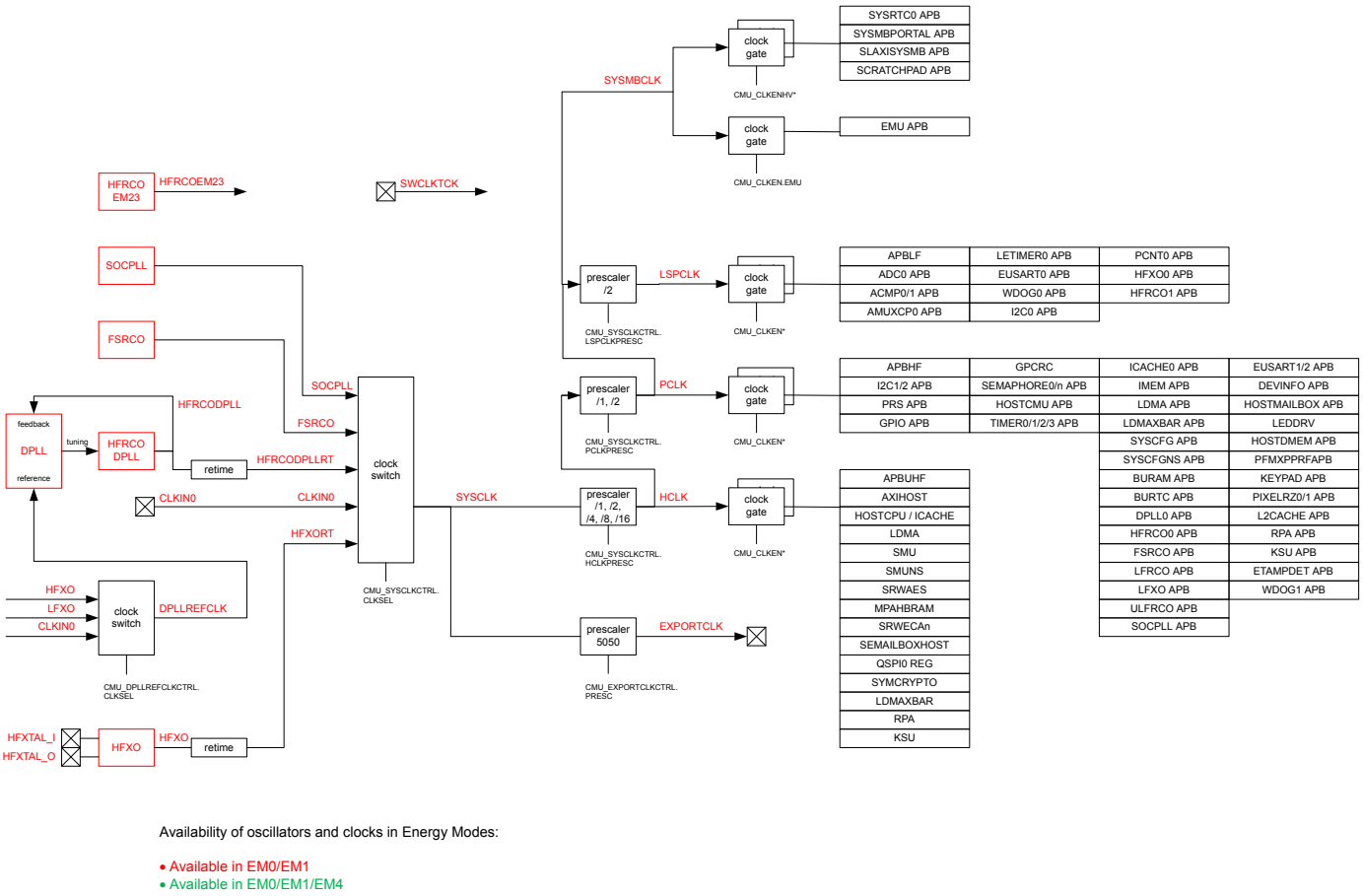


Figure 8.1. Bus Clocks

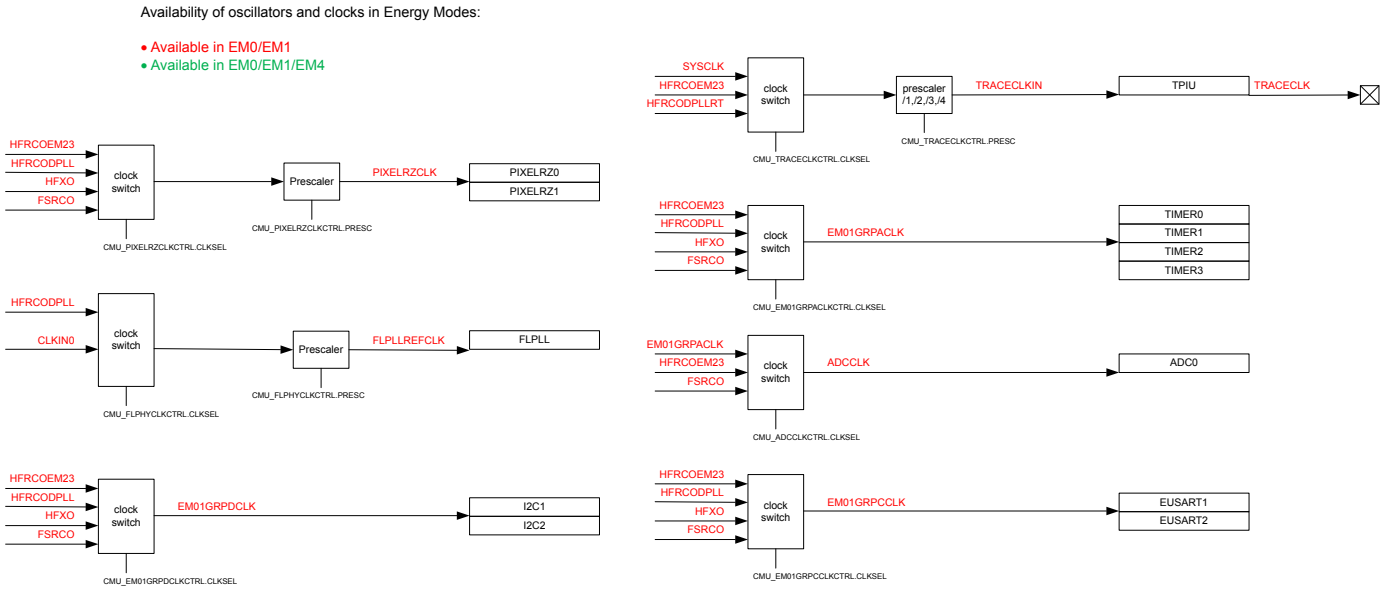


Figure 8.2. High-Frequency Peripheral Clocks

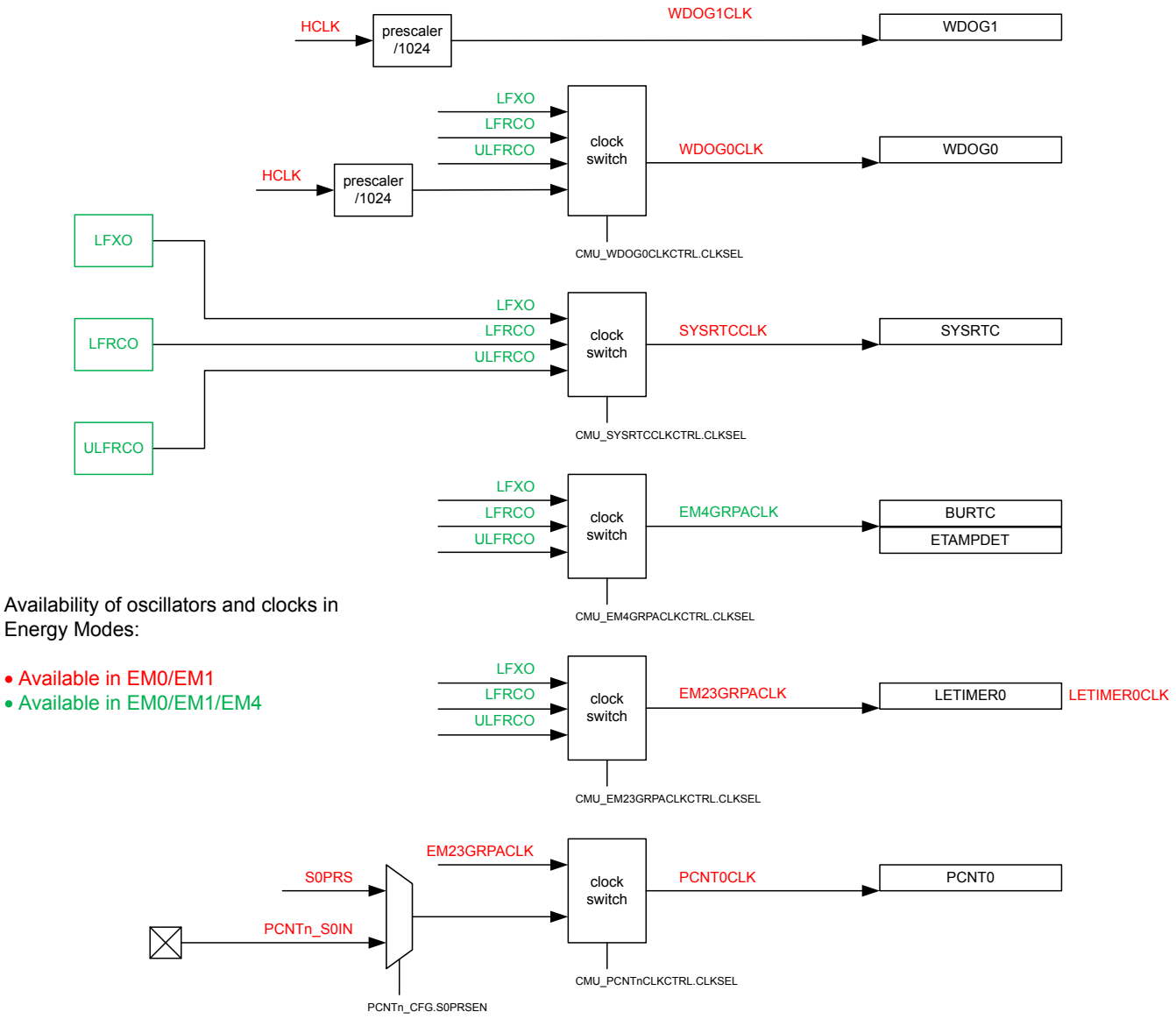


Figure 8.3. Low-Frequency Peripheral Clocks

Availability of oscillators and clocks in Energy Modes:

- Available in EM0/EM1
- Available in EM0/EM1/EM4

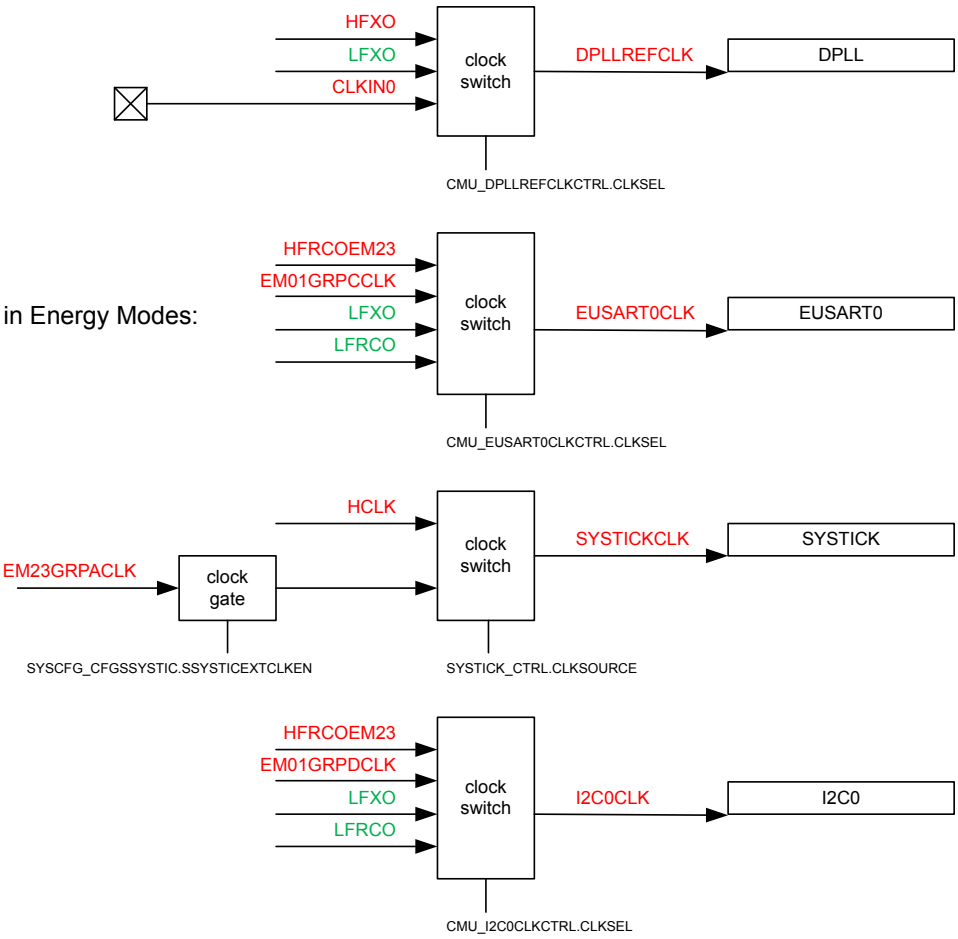


Figure 8.4. Mixed-Frequency Peripheral Clocks

8.3.1 System Clocks

8.3.1.1 SYSCLK - Bus Clock

SYSCLK is the selected System Clock. The SYSCLK can be driven by a high-frequency oscillator or be driven from a pin. HCLK, PCLKm LSPCLK and SYSMBCLK are derived from the SYSCLK tree. The system boots using the FSRCO oscillator, and switches to HFRCODPLL before user firmware execution begins. To change the selected clock source, write to the CLKSEL bitfield in CMU_SYSCLKCTRL. If an invalid option is programmed into CLKSEL, FSRCO will be selected. The SYSCLK is running in EM0 Active and EM1 Sleep and is automatically stopped in other energy modes.

The prescaler setting can be changed dynamically and the new setting takes effect immediately. When switching to a higher frequency oscillator source, prescaler settings for downstream branches should be adjusted before clock selection to prevent over clocking of any downstream branch. For the same reason, when switching to a lower frequency oscillator source, prescaler settings cannot be adjusted until the clock selection is made.

8.3.1.2 HCLK - AHB Clock

HCLK is a prescaled version of SYSCLK. This clock drives the AHB bus interface. HCLK can be prescaled by setting HCLKPRESC in CMU_SYSCLKCTRL from DIV2 to DIV16. This prescales HCLK to all AHB bus clocks and is typically used to save energy in applications where the system is not required to run at the highest frequency. The setting can be changed dynamically and the new setting takes effect immediately. Some of the modules that are driven by this clock can be clock gated completely when not in use. This is done by clearing the module enable (EN) bit in the module's EN register.

8.3.1.3 PCLK - APB Clock

PCLK is a prescaled version of HCLK. This clock drives the APB bus interface. PCLK can be prescaled by setting PCLKPRESC in CMU_SYSCLKCTRL to DIV2. This prescales PCLK to all APB bus clocks and is necessary to prevent PCLK from exceeding the maximum frequency when HCLK is operated at above 80 MHz. The setting can be changed dynamically and the new setting takes effect immediately. Peripheral APB interfaces that are driven by this clock will be clock gated completely when not in use. This is done by clearing the module enable (EN) bit in the module's EN register. Note the minimum frequency of PCLK is 2 MHz.

8.3.1.4 LSPCLK - Low Speed APB Clock

LSPCLK is a prescaled version of PCLK. This clock drives the Low Speed APB bus interface. LSPCLK is always prescaled by a factor of two. This prescales LSPCLK to all Low Speed APB bus clocks. Peripheral APB interfaces that are driven by this clock will be clock gated completely when not in use. This is done by clearing the module enable (EN) bit in the module's EN register. Note the minimum frequency of LSPCLK is 1 MHz and the maximum is 40 MHz. These will not be violated if PCLK limits are observed.

8.3.1.5 LPWRADIOCLK - Radio Clock

The radio clock (LPWRADIOCLK) is the clock for the low-power wireless (LPW) radio sub-system. It is the root clock for all the peripheral clocks and bus clocks for peripherals inside the LPW subsystem. LPWRADIOCLK is enabled with the CMU_RADIOCLKCTRL.EN bit. The radio stack code provided by Silicon Labs will manage this bit and the clock trees in the LPW subsystem.

8.3.1.6 EM01GRPACLK - EM01 Group A Clock

EM01GRPACLK is the selected clock for the Group A Peripherals in a PD1 power domain. These are typically high clock frequency peripheral modules. There are several selectable sources for EM01GRPACLK: HFXO, HFRCODPLL, and FSRCO. In addition, the EM01GRPACLK can be disabled. The selection is configured using the CLKSEL field in CMU_EM01GRPACLKCTRL.

Each High Frequency Peripheral that is clocked by EM01GRPACLK may have its own prescaler setting and enable bit. The prescaler settings, if available, can be found in the peripheral's control registers. The enable bit can be found in the module's EN register.

8.3.1.7 EM01GRPCCLK - EM01 Group C Clock

EM01GRPCCLK is the selected clock for the Group C Peripherals in a PD1 power domain. These are typically high clock frequency peripheral modules. There are several selectable sources for EM01GRPCCLK: HFXO, HFRCODPLL, HFRCOEM23, FSRCO, HFRCODPLLRT, and HFXORT. In addition, the EM01GRPCCLK can be disabled. The selection is configured using the CLKSEL field in CMU_EM01GRPCCLKCTRL.

Each High Frequency Peripheral that is clocked by EM01GRPCCLK may have its own prescaler setting and enable bit. The prescaler settings, if available, can be found in the peripheral's control registers. The enable bit can be found in the module's EN register.

8.3.1.8 EM01GRPDCLK - EM01 Group D Clock

EM01GRPDCLK is the selected clock for the Group D Peripherals in a PD1 power domain. These are typically high clock frequency peripheral modules. There are several selectable sources for EM01GRPDCLK: HFXO, HFRCODPLL, HFRCOEM23, FSRCO, HFRCODPLLRT, and HFXORT. In addition, the EM01GRPDCLK can be disabled. The selection is configured using the CLKSEL field in CMU_EM01GRPDCLKCTRL.

Each High Frequency Peripheral that is clocked by EM01GRPDCLK may have its own prescaler setting and enable bit. The prescaler settings, if available, can be found in the peripheral's control registers. The enable bit can be found in the module's EN register.

8.3.1.9 EM23GRPACLK - EM23 Group A Clock

EM23GRPACLK is the selected clock for the Group A Peripherals in a PD0 power domain. These are typically low energy consumption peripheral modules. There are three selectable sources for EM23GRPACLK: LFRCO, LFXO and ULFRCO. In addition, the EM23GRPACLK can be disabled. The selection is configured using the CLKSEL field in CMU_EM23GRPACLKCTRL.

Each Low Energy Peripheral that is clocked by EM23GRPACLK may have its own prescaler setting and enable bit. The prescaler settings, if available, can be found in the peripheral's control registers. The enable bit can be found in the module's EN register.

8.3.1.10 EM4GRPACLK - EM4 Group A Clock

EM4GRPACLK is the selected clock for the Group A Peripherals in the PDHV power domain. These are typically ultra low energy consumption peripheral modules. There are three selectable sources for EM4GRPACLK: LFRCO, LFXO and ULFRCO. In addition, the EM4GRPACLK can be disabled. The selection is configured using the CLKSEL field in CMU_EM4GRPACLKCTRL.

Note: EM4GRPACLK is in a different power domain than EM23GRPACLK, which makes it available all the way down to EM4.

Each Low Energy Peripheral that is clocked by EM4GRPACLK may have its own prescaler setting and enable bit. The prescaler settings, if available, can be found in the peripheral's control registers. The enable bit can be found in the module's EN register.

8.3.1.11 Peripheral Bus Clock Enable

Peripherals each have an individual bus clock enable bit in the CMU_CLKEN0 or CMU_CLKEN1 registers. Disabling the bus clock to a peripheral can save energy, even when that peripheral is not active.

8.3.1.12 ADCCLK - IADC Clock

ADCCLK is the selected clock for the ADC. The ADCCLK source may be selected from EM01GRPACLK, HFRCOEM23, or FSCRO. In addition, the ADCCLK can be disabled. The selection is configured using the CLKSEL field in CMU_ADCCLKCTRL.

Note: When using a Timer as the synchronous trigger for ADC conversion, EM01GRPACLK must be selected, because Timers run from EM01GRPACLK.

The ADC has its own prescaler setting and enable bit. The prescaler settings can be found in the ADC's control registers. The enable bit can be found in the ADC's EN register.

Whichever clock source is selected as the ADC clock via the CLKSEL bitfield in the CMU_ADCCLKCTRL register, this clock will become active automatically when needed. The ADC can automatically start and stop it.

8.3.1.13 EUSART0CLK - EUSART0 Clock

EUSART0CLK is the selected clock for the EUSART0 peripheral, and can choose between EM01GRPCCLK, HFRCOEM23, LFXO, and LFRCO. EUSART0CLK is selected via the CLKSEL field in CMU_EUSART0CLKCTRL. When operating EUSART0 as a high-speed UART or SPI main interface (EM0/1 only), EM01GRPACLK or HFRCOEM23 must be selected. To operate as a low-energy, low-speed peripheral, LFXO or LFRCO must be selected. To operate as a SPI secondary interface, EM01GRPCCLK or HFRCOEM23 should be selected.

8.3.1.14 I2C0CLK - I2C0 Clock

I2C0CLK is the selected clock for the I2C0 peripheral, and can choose between EM01GRPDCLK, HFRCOEM23, LFXO, and LFRCO. I2C0CLK is selected via the CLKSEL field in CMU_I2C0CLKCTRL. When operating I2C0CLK as a leader, EM01GRPDCLK or HFRCOEM23 must be selected to support the clock rate. To operate as a follower, LFXO or LFRCO may be selected.

8.3.1.15 PIXELRZCLK - PIXELRZ Clock

PIXELRZCLK is the selected clock for the PIXELRZ peripherals, and can choose between HFRCODPLL, HFRCOEM23, HFXO, and FSRCO. PIXELRZCLK is selected via the CLKSEL field in CMU_PIXELRZCLKCTRL, and an optional prescaler is supplied in the PRESC field. The PIXELRZCLK has an upper limit of 40 MHz.

8.3.1.16 TRACECLK - Debug Trace Clock

The TRACECLK is the clock provided to the ETM trace feature and is limited to 40 MHz maximum. By default it is derived from SYSCLK, but can be changed to source from HFRCODPLLRT or HFRCOEM23 using the CLKSEL field in CMU_TRACECLKCTRL. The selected clock source is scaled via the PRESC field in the CMU_TRACECLKCTRL register. For example, if the selected clock is SYSCLK running at 40 MHz or less, the default DIV1 setting may be used. When the selected clock is above 40 MHz, use higher divider values to avoid data pump overflow. The selected debug trace clock will be used to run the Cortex-M33 trace logic. Note that this register should be configured properly before enabling ETM.

8.3.1.17 WDOG0CLK - Watchdog Timer 0 Clock

The Watchdog Timer 0 (WDOG0) can be configured to use one of four different clock sources: LFRCO, LFXO, ULFRCO, or HCLKDIV1024. Select option HCLKDIV1024 to track Watchdog timeout with CPU clock speed.

8.3.1.18 WDOG1CLK - Watchdog Timer 1 Clock

The Watchdog Timer 1 (WDOG1) is always configured to run at HCLKDIV1024.

8.3.2 Switching Clock Source

The FSRCO oscillator is a fixed frequency (20 MHz), low energy oscillator with extremely short start-up time. Therefore, this oscillator is chosen by hardware as the clock source for SYSCLK when the device starts up (e.g. after reset).

Software can switch between the different clock sources at run-time. For example, when the HFRCODPLL is the clock source, software can switch to HFXO by writing the field CLKSEL in the CMU_SYSCLKCTRL register. See Figure 8.5. CMU Switching From HFRCO to HFXO Before HFXO is Ready for a description of the sequence of events for this specific operation.

When switching the SYSCLK to HFXO via the CLKSEL bitfield in CMU_SYSCLKCTRL, HFXO is automatically started. Switching to an oscillator that is not ready yet, the SYSCLK will stop for the duration of the oscillator start-up time. This effectively stalls the Core Modules. It is possible to avoid this by first enabling the target oscillator (e.g. HFXO) and then waiting for that oscillator to become ready before switching the clock source. This way, the system continues to run on the HFRCO until the target oscillator (e.g. HFXO) is ready and provides a reliable clock. This sequence of events is shown in Figure 8.6. CMU Switching From HFRCO to HFXO After HFXO is Ready.

Generally, all oscillators have a separate flag that is set when the oscillator is ready. This flag can also be configured to generate an interrupt.

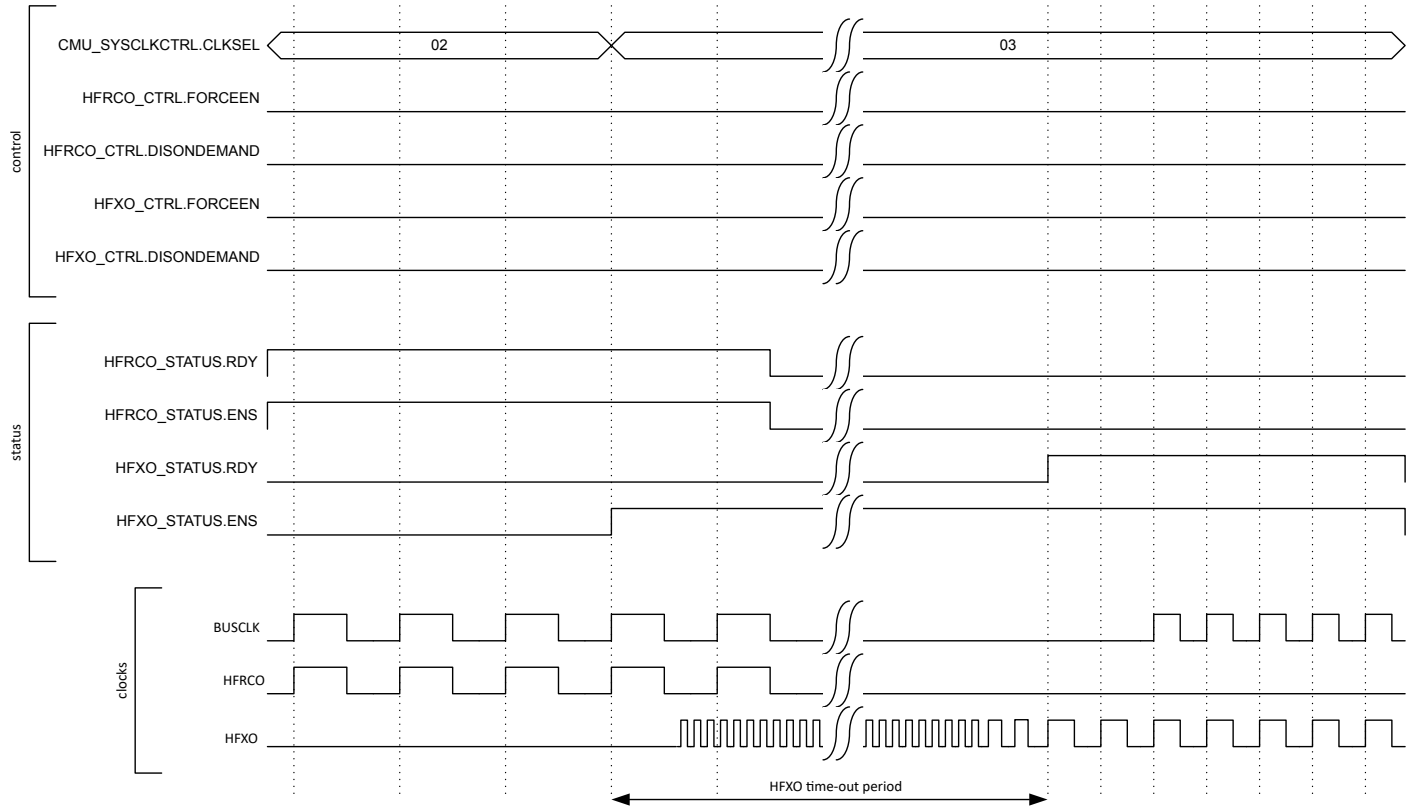


Figure 8.5. CMU Switching From HFRCO to HFXO Before HFXO is Ready

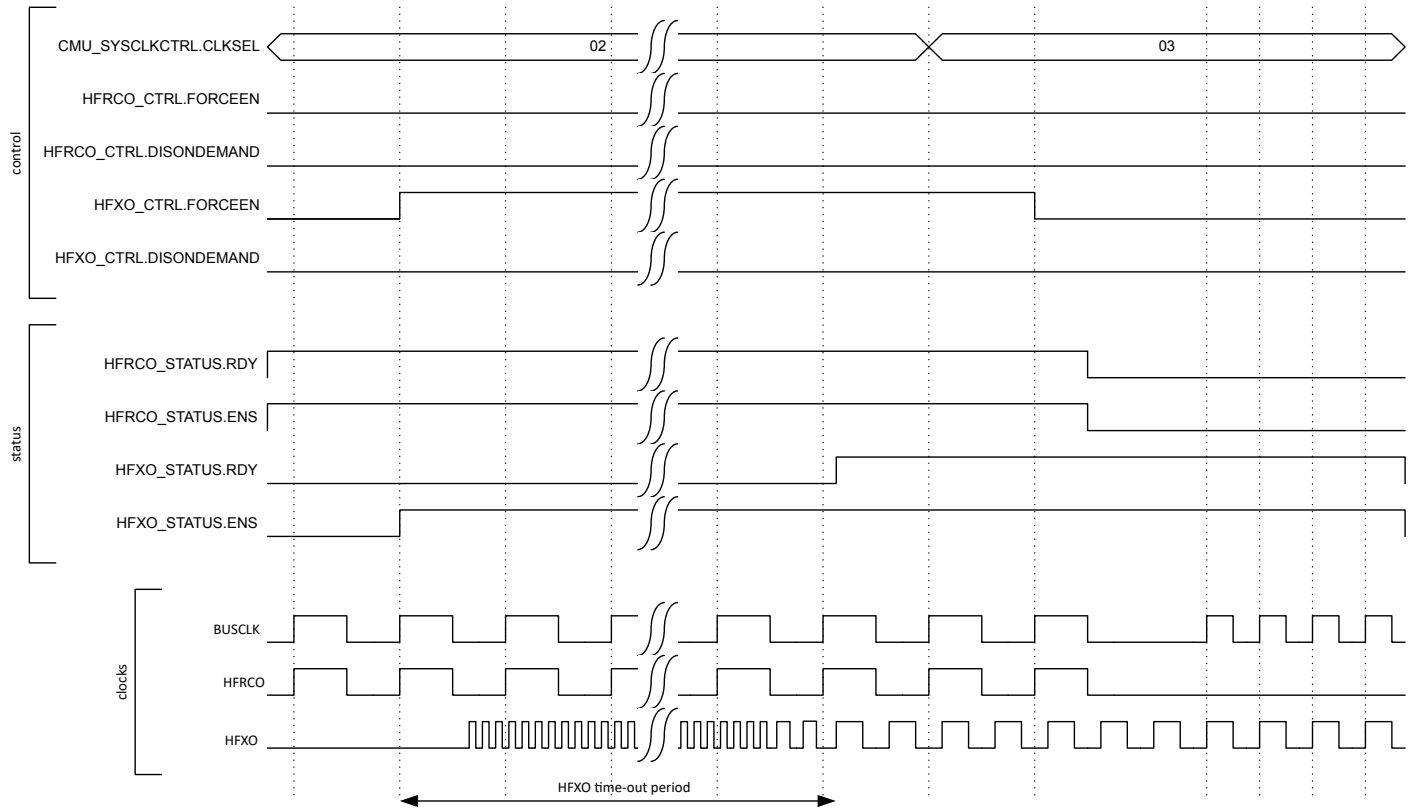


Figure 8.6. CMU Switching From HFRCO to HFXO After HFXO is Ready

Switching clock source for various clock switches is done by setting the CLKSEL bitfields in CMU_*CLKCTRL. To ensure no stalls in the peripherals, the clock source should be ready before switching to it.

Note: To save energy, remember to disable all clock switches and/or module enable bits when not in use.

8.3.3 Oscillator Calibration and Measurement

The CMU has built-in hardware support to efficiently calibrate RC oscillators (HFRCO0 / HFRCODPLL, HFRCOEM23) at run-time or measure the timing of other oscillators or periodic signals routed via PRS, see Figure 8.7. Hardware Support for RC Oscillator Calibration for an illustration of this circuit.

The concept is to select a reference frequency and compare another frequency to the reference. When the calibration circuit is started, one down-counter running on a selectable clock (DOWNSEL in CMU_CALCTRL) and one up-counter running on a selectable clock (UPSEL in CMU_CALCTRL) are started simultaneously. Reference clocks may also be routed through the PRS channels via the CALUP and CALDN consumer inputs. The top value for the down-counter must be written to the CMU_CALTOP register before calibration is started. The down-counter counts for CALTOP + 1 cycles. When the down-counter has reached 0, the up-counter is sampled and the CALRDY interrupt flag in the IF register is set. If CONT in CMU_CALCTRL is cleared, the counters are stopped after finishing the ongoing calibration. If continuous mode is selected by setting CONT in CMU_CALCTRL, the down-counter reloads the top value and continues counting, while the up-counter restarts from 0.

When WRAPUP in CMU_CALCTRL is cleared to '0', overflows of the up-counter will not occur. If the up-counter reaches its maximum value before the down-counter reaches 0, the up-counter stays at its top value. If the up-counter is saturated at its maximum value in this case, the measurement interval was too long to hold in the up-counter, and a shorter interval should be used. When the down-counter has reached 0, software can read out the sampled up-counter value from CMU_CALCNT. The up-counter has counted (the sampled value) + 1 cycles. The ratio between the reference and the oscillator subject to the measurement or calibration can easily be found using (the top value) +1 and (the sampled value) +1.

If WRAPUP is set to '1' and the up-counter reaches its maximum value, the up-counter will wrap around and continue counting again from zero instead of saturating. This allows longer time intervals to be measured with additional precision. However, to use this feature the expected frequency range of the measured value must be known. When WRAPUP is set to '1' it is recommended to select the down-count interval to ensure that a known number of up-count overflows will occur during this interval, and the extremes of the up-count frequency will not cause the result to be near an up-count overflow event. When the down-counter reaches 0 using this method, software can read out the sampled up-counter value from CMU_CALCNT. In this case, the up-counter has counted (up-counter range * overflows) + (the sampled value) + 1 cycles.

Calibration can be started and stopped by writing CALSTART and CALSTOP bitfields in CMU_CALCMD, respectively. With this hardware support, it is simple to write efficient software calibration algorithms.

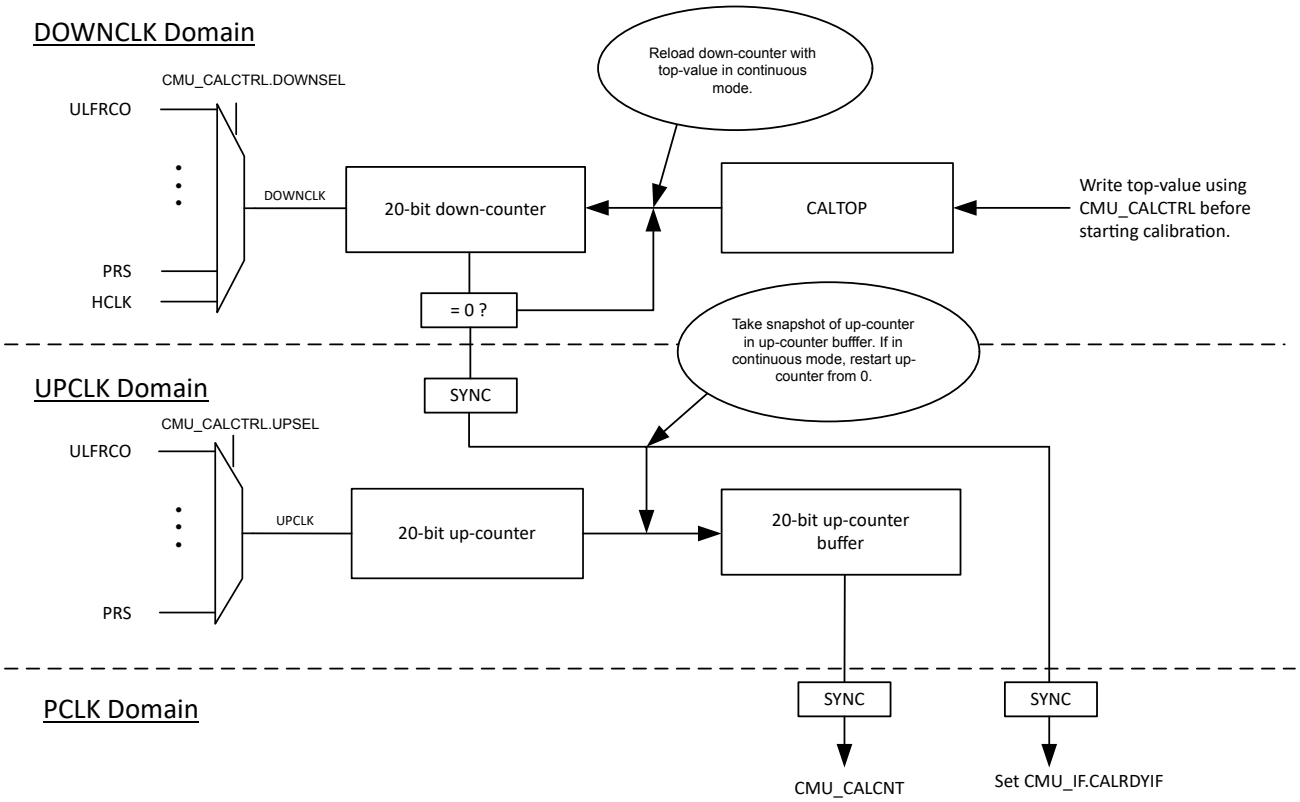


Figure 8.7. Hardware Support for RC Oscillator Calibration

The counter operation for single and continuous mode are shown in Figure 8.8. Single Calibration (CONT=0) and Figure 8.9. Continuous Calibration (CONT=1) respectively.

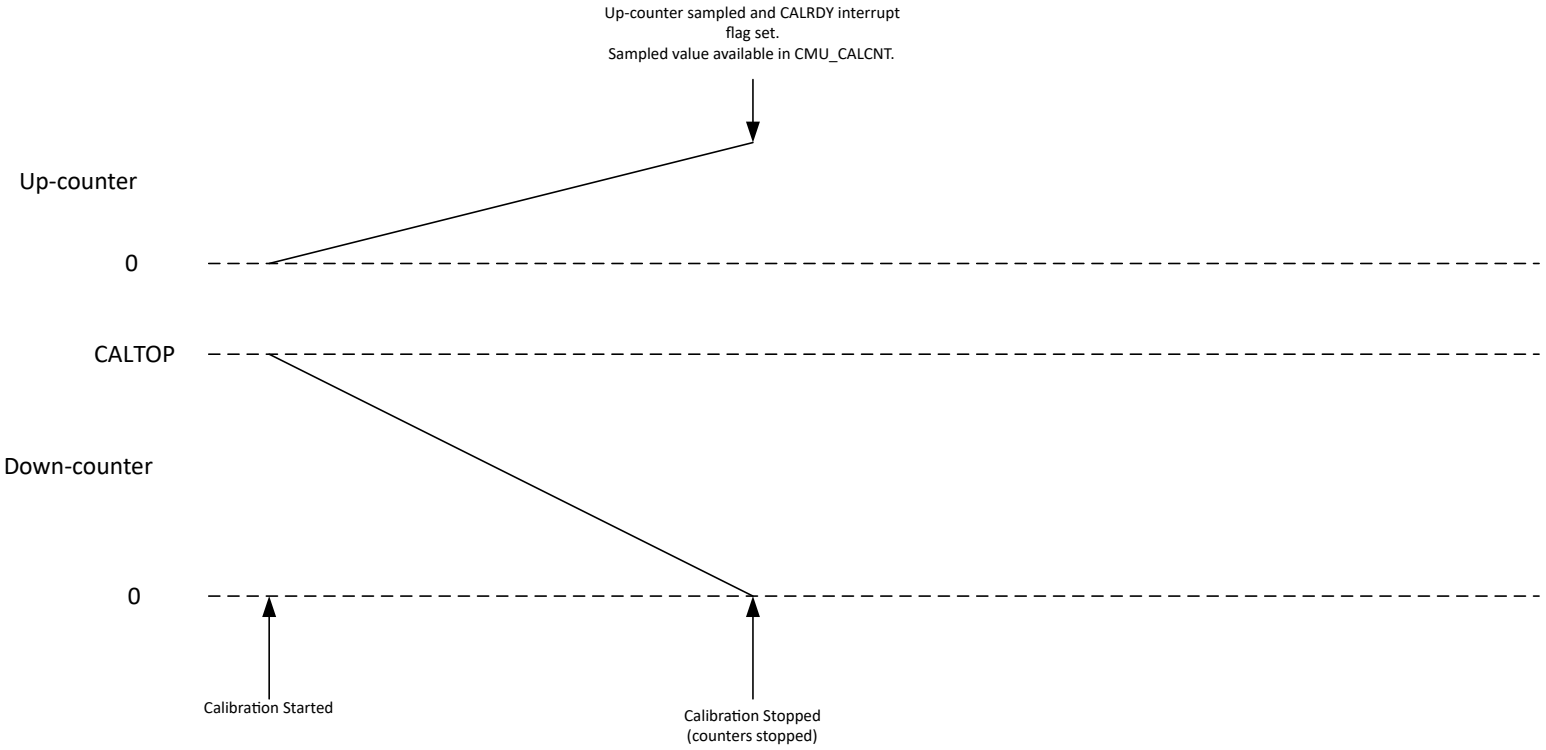


Figure 8.8. Single Calibration (CONT=0)

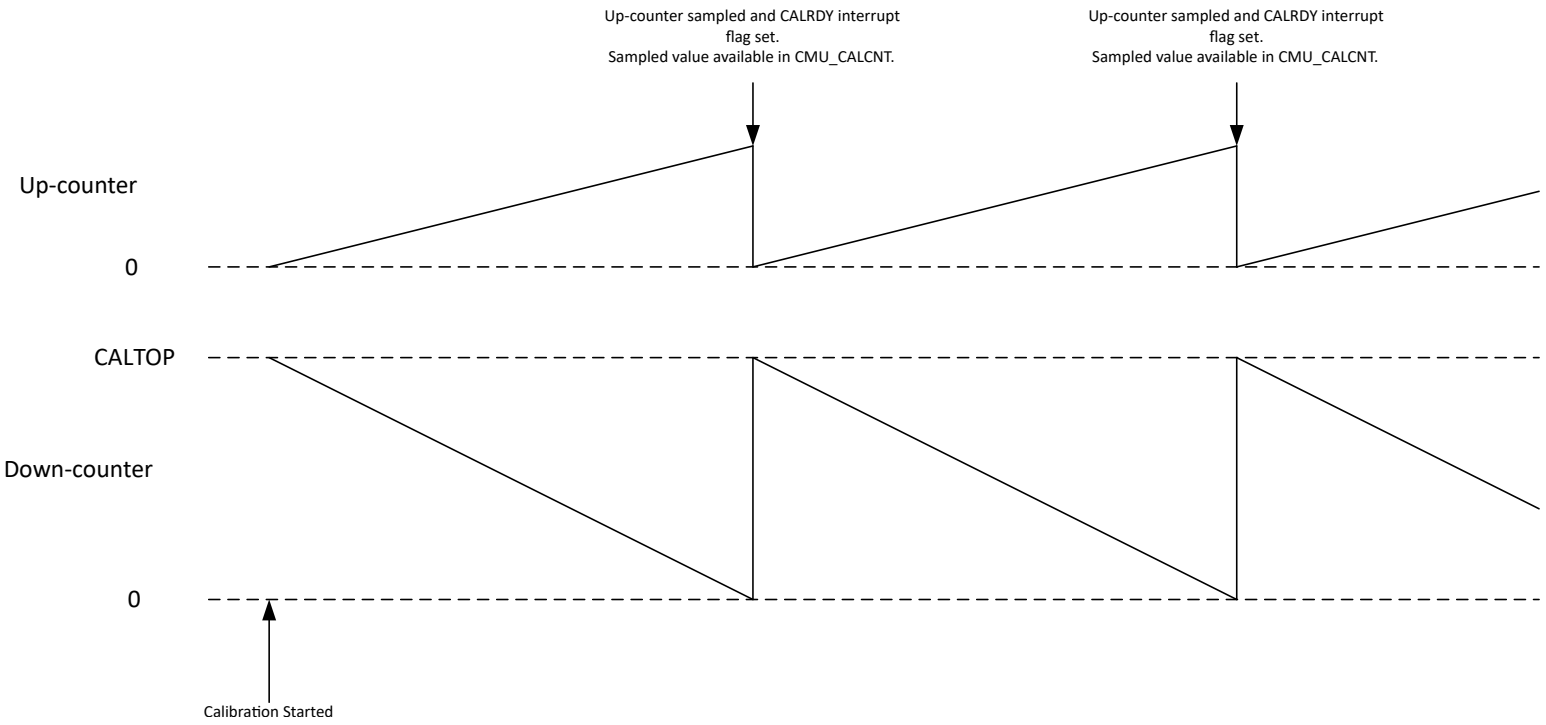


Figure 8.9. Continuous Calibration (CONT=1)

8.3.4 Energy Modes

The availability of oscillators and system clocks depends on the chosen energy mode. By default, all clocks are available down to EM1 Sleep. In EM4 the high frequency oscillators and precicion mode of the LFRCO are not available, but low-frequency oscillators may still be used to time sleep intervals. Clock availability is summarized in Table 8.1. Oscillator and Clock Availability in Energy Modes. The CMU figures in 8.3 Functional Description also indicate which oscillators and clocks can be used in what energy modes.

By default, the low frequency oscillators (LFRCO, LFXO,and ULFRCO) off in EM4 Shutoff. The LFXO or LFRCO (in non-precision mode) can be requested in EM4 as needed. When the LFRCO precision mode is used, this oscillator cannot operate in EM4. The low frequency clock trees (EM23GRPACLK, EM4GRPACLK, WDOGCLK, RTCCCLK, and PRORTCCLK) are in various power domains and their availability depends both on the chosen clock source and on the energy mode as indicated in Table 8.1. Oscillator and Clock Availability in Energy Modes.

Table 8.1. Oscillator and Clock Availability in Energy Modes

	EM0 Active	EM1 Sleep	EM4 Shutoff
SOCPLL	On -Demand[1]	On -Demand[1]	Off
HFRCO0 / HFRCODPLL	On -Demand[1]	On -Demand[1]	Off
HFRCOEM23	On -Demand[1]	On -Demand[1]	Off
HFXO	On -Demand[1]	On -Demand[1]	Off
FSRCO	On -Demand[1]	On -Demand[1]	Off
LFRCO, LFXO	On -Demand[1]	On -Demand[1]	On -Demand [2]
ULFRCO	On	On	On -Demand[3]
SYSCLK, HCLK, PCLK, LSPCLK, SYSMBCLK	On -Demand[3]	On -Demand[3]	Off
ADCCLK, EM01GRPACLK, EM01GRPCCLK, EM23GRPACLK, EUSART0CLK, FLPLLREFCLK, I2C0CLK, PCNTCLK, PIX-ELRZCLK, SYSRTCCLK, SYSTICKCLK, WDOG0CLK, WDOG1CLK	On -Demand[3]	On -Demand[3]	Off
EM4GRPACLK	On -Demand[3]	On -Demand[3]	On -Demand[3]

1 Can be enabled by software or hardware.
2 LFRCO not available in EM4 when precision mode enabled.
3 Enabled or disabled by hardware only.

8.3.5 Clock Output

The CMU has up to three CLKOUTn signals that can be routed to the PRS or GPIO. The selections for CLKOUTn are controlled using the CLKOUTSELn bitfields in CMU_EXPORTCLKCTRL (CLKOUTSEL0 controls CLKOUT0, for example).

The following clocks can be selected for CLKOUTn:

- HCLK and EXPORTCLK. The HCLK is the high frequency clock for AHB. The EXPORTCLK is a prescaled version of SYSCLK as controlled by the PRESC bitfield in the CMU_EXPORTCLKCTRL register.
- The qualified clock from any of the on-chip oscillators. A qualified clock will not have any glitches or skewed duty-cycle during start-up. For the LFXO and HFXO, correct configuration of the TIMEOUT bitfield(s) in LFXO_CFG and HFXO_XTALCFG, respectively is required to guarantee a properly qualified clock.

HCLK will only have a 50-50 duty cycle when HCLKPRESC in CMU_SYSCLKCTRL is DIV1. EXPORTCLK will only be 50-50 duty cycle when the selected division factor is even.

The CLKOUTn signals may be routed to GPIO via the DBUS as CMU.CLKOUTn using controls in the GPIO registers. The required output pins must be enabled in the GPIO_CMU_ROUTEEN register and the pin locations can be configured in the GPIO_CMU_CLKOUTnROUTE registers.

The CLKOUTn signals can also be used as PRS producers (see 14.3.3 Producers for more detail on PRS producers). CLKOUTn signals used as PRS producers may be simultaneously routed to GPIO, but this is not required to use CLKOUTn as a PRS producer.

8.3.6 Clock Input from a Pin

It is possible to configure the CMU to input a clock from CMU.CLKIN0. This clock can be selected to drive SYSCLK and DPLL reference using CMU_SYSCLKCTRL.CLKSEL and CMU_DPLLREFCLKCTRL.CLKSEL respectively. The required input pin locations can be configured in the GPIO_CMU_CLKIN0ROUTE register.

8.3.7 Interrupts

The interrupts generated by the CMU module are combined into one interrupt vector. If CMU interrupts are enabled, an interrupt will be made if one or more of the interrupt flags in CMU_IF and their corresponding bits in CMU_IEN are set.

8.3.8 Protection

It is possible to lock the control and command registers to prevent unintended software writes to critical clock settings. This is controlled by the CMU_LOCK register.

The WDOGCLKCTRL registers are separately locked by CMU_WDOGLOCK register.

In addition to software locks, hardware locks are implemented to prevent metastability. CMU_CALCTRL is locked by hardware when calibration is started by CMU_CALCMD.CALSTART. CMU_DPLLREFCLKCTRL is locked by hardware when DPLL is enabled via DPLL_EN.EN. Because these switches are not glitch-less, clock selection must be configured before enabling the operation and cannot be changed during operation.

8.4 CMU Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	CMU_IPVERSION	R	IP Version ID
0x008	CMU_STATUS	RH	Status Register
0x010	CMU_LOCK	W	Configuration Lock Register
0x014	CMU_WDOGLOCK	W	WDOG Configuration Lock Register
0x020	CMU_IF	RWH INTFLAG	Interrupt Flag Register
0x024	CMU_IEN	RW	Interrupt Enable Register
0x050	CMU_CALCMD	W	Calibration Command Register
0x054	CMU_CALCTRL	RW	Calibration Control Register
0x058	CMU_CALCNT	R	Calibration Result Counter Register
0x05C	CMU_CALTOP	RW	Calibration Top Value Register
0x064	CMU_CLKEN0	RW	Clock Enable Register 0
0x068	CMU_CLKEN1	RW	Clock Enable Register 1
0x070	CMU_SYSCLKCTRL	RW	System Clock Control
0x080	CMU_TRACECLKCTRL	RW	Debug Trace Clock Control
0x090	CMU_EXPORTCLKCTRL	RW	Export Clock Control
0x100	CMU_DPLLREFCLKCTRL	RW	Digital PLL Reference Clock Control
0x120	CMU_EM01GRPACLKCTRL	RW	EM01 Peripheral Group A Clock Control
0x128	CMU_EM01GRPCCLKCTRL	RW	EM01 Peripheral Group C Clock Control
0x140	CMU_EM23GRPACLKCTRL	RW	EM23 Peripheral Group A Clock Control
0x160	CMU_EM4GRPACLKCTRL	RW	EM4 Peripheral Group A Clock Control
0x180	CMU_ADCCLKCTRL	RW	ADC Clock Control
0x200	CMU_WDOG0CLKCTRL	RW	WDOG0 Clock Control
0x220	CMU_EUSART0CLKCTRL	RW	EUSART0 Clock Control
0x240	CMU_SYSRTC0CLKCTRL	RW	SYSRTC0 Clock Control
0x270	CMU_PCNT0CLKCTRL	RW	PCNT0 Clock Control
0x280	CMU_RADIOCLKCTRL	RW	Radio Clock Control
0x29C	CMU_HFXO0LFCLKCTRL	RW	HFXO0 LF Clock Control
0x2A8	CMU_EM01GRPDCLKCTRL	RW	EM01 Peripheral Group D Clock Control
0x2AC	CMU_I2C0CLKCTRL	RW	I2C0 Clock Control
0x2B0	CMU_PIXELRZCLKCTRL	RW	PIXELRZ Clock Control
0x2B4	CMU_CLKENHV	RW	Clock Enable Register 2
0x1000	CMU_IPVERSION_SET	R	IP Version ID
0x1008	CMU_STATUS_SET	RH	Status Register
0x1010	CMU_LOCK_SET	W	Configuration Lock Register
0x1014	CMU_WDOGLOCK_SET	W	WDOG Configuration Lock Register
0x1020	CMU_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1024	CMU_IEN_SET	RW	Interrupt Enable Register
0x1050	CMU_CALCMD_SET	W	Calibration Command Register
0x1054	CMU_CALCTRL_SET	RW	Calibration Control Register

Offset	Name	Type	Description
0x1058	CMU_CALCNT_SET	R	Calibration Result Counter Register
0x105C	CMU_CALTOP_SET	RW	Calibration Top Value Register
0x1064	CMU_CLKEN0_SET	RW	Clock Enable Register 0
0x1068	CMU_CLKEN1_SET	RW	Clock Enable Register 1
0x1070	CMU_SYSCLKCTRL_SET	RW	System Clock Control
0x1080	CMU_TRACECLKCTRL_SET	RW	Debug Trace Clock Control
0x1090	CMU_EXPORTCLKCTRL_SET	RW	Export Clock Control
0x1100	CMU_DPLLREFCLKCTRL_SET	RW	Digital PLL Reference Clock Control
0x1120	CMU_EM01GRPACLKCTRL_SET	RW	EM01 Peripheral Group A Clock Control
0x1128	CMU_EM01GRPCCLKCTRL_SET	RW	EM01 Peripheral Group C Clock Control
0x1140	CMU_EM23GRPACLKCTRL_SET	RW	EM23 Peripheral Group A Clock Control
0x1160	CMU_EM4GRPACLKCTRL_SET	RW	EM4 Peripheral Group A Clock Control
0x1180	CMU_ADCCLKCTRL_SET	RW	ADC Clock Control
0x1200	CMU_WDOG0CLKCTRL_SET	RW	WDOG0 Clock Control
0x1220	CMU_EUSART0CLKCTRL_SET	RW	EUSART0 Clock Control
0x1240	CMU_SYSRTC0CLKCTRL_SET	RW	SYSRTC0 Clock Control
0x1270	CMU_PCNT0CLKCTRL_SET	RW	PCNT0 Clock Control
0x1280	CMU_RADIOCLKCTRL_SET	RW	Radio Clock Control
0x129C	CMU_HFXO0LFCLKCTRL_SET	RW	HFXO0 LF Clock Control
0x12A8	CMU_EM01GRPDCLKCTRL_SET	RW	EM01 Peripheral Group D Clock Control
0x12AC	CMU_I2C0CLKCTRL_SET	RW	I2C0 Clock Control
0x12B0	CMU_PIXELRZCLKCTRL_SET	RW	PIXELRZ Clock Control
0x12B4	CMU_CLKENHV_SET	RW	Clock Enable Register 2
0x2000	CMU_IPVERSION_CLR	R	IP Version ID
0x2008	CMU_STATUS_CLR	RH	Status Register
0x2010	CMU_LOCK_CLR	W	Configuration Lock Register
0x2014	CMU_WDOGLOCK_CLR	W	WDOG Configuration Lock Register
0x2020	CMU_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2024	CMU_IEN_CLR	RW	Interrupt Enable Register
0x2050	CMU_CALCMD_CLR	W	Calibration Command Register
0x2054	CMU_CALCTRL_CLR	RW	Calibration Control Register
0x2058	CMU_CALCNT_CLR	R	Calibration Result Counter Register
0x205C	CMU_CALTOP_CLR	RW	Calibration Top Value Register
0x2064	CMU_CLKEN0_CLR	RW	Clock Enable Register 0
0x2068	CMU_CLKEN1_CLR	RW	Clock Enable Register 1
0x2070	CMU_SYSCLKCTRL_CLR	RW	System Clock Control
0x2080	CMU_TRACECLKCTRL_CLR	RW	Debug Trace Clock Control
0x2090	CMU_EXPORTCLKCTRL_CLR	RW	Export Clock Control
0x2100	CMU_DPLLREFCLKCTRL_CLR	RW	Digital PLL Reference Clock Control

Offset	Name	Type	Description
0x2120	CMU_EM01GRPACLKCTRL_CLR	RW	EM01 Peripheral Group A Clock Control
0x2128	CMU_EM01GRPCCLKCTRL_CLR	RW	EM01 Peripheral Group C Clock Control
0x2140	CMU_EM23GRPACLKCTRL_CLR	RW	EM23 Peripheral Group A Clock Control
0x2160	CMU_EM4GRPACLKCTRL_CLR	RW	EM4 Peripheral Group A Clock Control
0x2180	CMU_ADCCLKCTRL_CLR	RW	ADC Clock Control
0x2200	CMU_WDOG0CLKCTRL_CLR	RW	WDOG0 Clock Control
0x2220	CMU_EUSART0CLKCTRL_CLR	RW	EUSART0 Clock Control
0x2240	CMU_SYSRTC0CLKCTRL_CLR	RW	SYSRTC0 Clock Control
0x2270	CMU_PCNT0CLKCTRL_CLR	RW	PCNT0 Clock Control
0x2280	CMU_RADIOCLKCTRL_CLR	RW	Radio Clock Control
0x229C	CMU_HFXO0LFCLKCTRL_CLR	RW	HFXO0 LF Clock Control
0x22A8	CMU_EM01GRPDCLKCTRL_CLR	RW	EM01 Peripheral Group D Clock Control
0x22AC	CMU_I2C0CLKCTRL_CLR	RW	I2C0 Clock Control
0x22B0	CMU_PIXELRZCLKCTRL_CLR	RW	PIXELRZ Clock Control
0x22B4	CMU_CLKENHV_CLR	RW	Clock Enable Register 2
0x3000	CMU_IPVERSION_TGL	R	IP Version ID
0x3008	CMU_STATUS_TGL	RH	Status Register
0x3010	CMU_LOCK_TGL	W	Configuration Lock Register
0x3014	CMU_WDOGLOCK_TGL	W	WDOG Configuration Lock Register
0x3020	CMU_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3024	CMU_IEN_TGL	RW	Interrupt Enable Register
0x3050	CMU_CALCMD_TGL	W	Calibration Command Register
0x3054	CMU_CALCTRL_TGL	RW	Calibration Control Register
0x3058	CMU_CALCNT_TGL	R	Calibration Result Counter Register
0x305C	CMU_CALTOP_TGL	RW	Calibration Top Value Register
0x3064	CMU_CLKEN0_TGL	RW	Clock Enable Register 0
0x3068	CMU_CLKEN1_TGL	RW	Clock Enable Register 1
0x3070	CMU_SYSCLKCTRL_TGL	RW	System Clock Control
0x3080	CMU_TRACECLKCTRL_TGL	RW	Debug Trace Clock Control
0x3090	CMU_EXPORTCLKCTRL_TGL	RW	Export Clock Control
0x3100	CMU_DPLLREFCLKCTRL_TGL	RW	Digital PLL Reference Clock Control
0x3120	CMU_EM01GRPACLKCTRL_TGL	RW	EM01 Peripheral Group A Clock Control
0x3128	CMU_EM01GRPCCLKCTRL_TGL	RW	EM01 Peripheral Group C Clock Control
0x3140	CMU_EM23GRPACLKCTRL_TGL	RW	EM23 Peripheral Group A Clock Control
0x3160	CMU_EM4GRPACLKCTRL_TGL	RW	EM4 Peripheral Group A Clock Control
0x3180	CMU_ADCCLKCTRL_TGL	RW	ADC Clock Control
0x3200	CMU_WDOG0CLKCTRL_TGL	RW	WDOG0 Clock Control

Offset	Name	Type	Description
0x3220	CMU_EUSART0CLKCTRL_TGL	RW	EUSART0 Clock Control
0x3240	CMU_SYSRTC0CLKCTRL_TGL	RW	SYSRTC0 Clock Control
0x3270	CMU_PCNT0CLKCTRL_TGL	RW	PCNT0 Clock Control
0x3280	CMU_RADIOCLKCTRL_TGL	RW	Radio Clock Control
0x329C	CMU_HFXO0LFCLKCTRL_TGL	RW	HFXO0 LF Clock Control
0x32A8	CMU_EM01GRPDCLKCTRL_TGL	RW	EM01 Peripheral Group D Clock Control
0x32AC	CMU_I2C0CLKCTRL_TGL	RW	I2C0 Clock Control
0x32B0	CMU_PIXELRZCLKCTRL_TGL	RW	PIXELRZ Clock Control
0x32B4	CMU_CLKENHV_TGL	RW	Clock Enable Register 2

8.5 CMU Register Description

8.5.1 CMU_IPVERSION - IP Version ID

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x8																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x8	R	**IP Version ID**

8.5.2 CMU_STATUS - Status Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0																														0x0
Access	R	R																														R
Name	LOCK	WDOGLOCK																														CALRDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**Configuration Lock Status**
	Indicates the current status of configuration lock			

Value	Mode	Description
0	UNLOCKED	Configuration lock is unlocked
1	LOCKED	Configuration lock is locked

Bit	Name	Reset	Access	Description
30	WDOGLOCK	0x0	R	**Configuration Lock Status for WDOG**
	Indicates the current status of WDOG configuration lock			

Value	Mode	Description
0	UNLOCKED	WDOG configuration lock is unlocked
1	LOCKED	WDOG configuration lock is locked

Bit	Name	Reset	Access	Description
29:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CALRDY	0x0	R	**Calibration Ready**
	Calibration is Ready (0 when calibration is ongoing).			

8.5.3 CMU_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x93F7															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x93F7	W	**Configuration Lock Key**

Write any other value than the unlock code to lock registers from editing. Write the unlock code to unlock.

Value	Mode	Description
37879	UNLOCK	Write this value to unlock

8.5.4 CMU_WDOGLOCK - WDOG Configuration Lock Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x5257															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x5257	W	**Configuration Lock Key**

Write any other value than the unlock code to lock registers from editing. Write the unlock code to unlock.

Value	Mode	Description
37879	UNLOCK	Write this value to unlock

8.5.5 CMU_IF - Interrupt Flag Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															CALOF	CALRDY

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CALOF	0x0	RW	**Calibration Overflow Interrupt Flag**
	Set when calibration overflow has occurred (i.e. if a new calibration completes before CMU_CALSTATUS has been read)			
0	CALRDY	0x0	RW	**Calibration Ready Interrupt Flag**
	Set when calibration is completed			

8.5.6 CMU_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															CALOF	CALRDY

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CALOF	0x0	RW	**Calibration Overflow Interrupt Enable**
	Enable/disable CALOF interrupt			
0	CALRDY	0x0	RW	**Calibration Ready Interrupt Enable**
	Enable/disable CALRDY interrupt			

8.5.7 CMU_CALCMD - Calibration Command Register

Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															W(nB)	W(nB)
Name																															CALSTOP	CALSTART

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CALSTOP	0x0	W(nB)	**Calibration Stop**
	Stops the calibration counters.			
0	CALSTART	0x0	W(nB)	**Calibration Start**
	Starts the calibration, effectively loading the CMU_CALCTRL.CALCNT into the down-counter and start decrementing.			

8.5.8 CMU_CALCTRL - Calibration Control Register

Offset	Bit Position																													
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset												0x0								0x0								0x0				0x0
Access												RW								RW								RW				RW
Name												DOWNSEL								UPSEL								CONT				WRAPUP

Bit	Name	Reset	Access	Description
31:21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20:16	DOWNSEL	0x0	RW	**Calibration Down-counter Select**

Selects clock source for the calibration down-counter. Changing this while calibration is running results in bus fault..

Value	Mode	Description
0	DISABLED	Down-counter is not clocked
1	HCLK	HCLK is clocking down-counter
2	PRS	PRS CMU_CALDN consumer is clocking down-counter
3	HFXO	HFXO is clocking down-counter
4	LFXO	LFXO is clocking down-counter
5	HFRCODPLL	HFRCODPLL is clocking down-counter
6	HFRCOEM23	HFRCOEM23 is clocking down-counter
7	FSRCO	FSRCO is clocking down-counter
8	LFRCO	LFRCO is clocking down-counter
9	ULFRCO	ULFRCO is clocking down-counter
10	HFRCOLPW	HFRCOLPW is clocking down-counter
11	SOCPLL	SOCPLL is clocking down-counter

15:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12:8	UPSEL	0x0	RW	**Calibration Up-counter Select**

Selects clock source for the calibration up-counter. Changing this while calibration is running results in bus fault.

Value	Mode	Description
0	DISABLED	Up-counter is not clocked
1	HCLK	HCLK is clocking up-counter
2	PRS	PRS is clocking down-counter
3	HFXO	HFXO is clocking up-counter
4	LFXO	LFXO is clocking up-counter
5	HFRCODPLL	HFRCODPLL is clocking up-counter
6	HFRCOEM23	HFRCOEM23 is clocking up-counter
7	FSRCO	FSRCO is clocking up-counter
8	LFRCO	LFRCO is clocking up-counter

Bit	Name	Reset	Access	Description
	9	ULFRCO		ULFRCO is clocking up-counter
	10	HFRCOLPW		HFRCOLPW is clocking up-counter
	11	SOCPLL		SOCPLL is clocking up-counter
7:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	CONT	0x0	RW	**Continuous Calibration**
	Set this bit to enable continuous calibration. Changing this while calibration is running results in bus fault.			
3:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	WRAPUP	0x0	RW	**Wrap around Up-counter**
	Set this bit to enable the up-counter to wrap around instead of saturating.			

8.5.9 CMU_CALCNT - Calibration Result Counter Register

Offset	Bit Position																													
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																					
Access									R																					
Name									CALCNT																					

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	CALCNT	0x0	R	**Calibration Result Counter Value**
	Read calibration result when Calibration Ready flag has been set.			

8.5.10 CMU_CALTOP - Calibration Top Value Register

Offset	Bit Position																													
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																					
Access									RW																					
Name									CALTOP																					

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	CALTOP	0x0	RW	**Calibration Counter Top Value**
	Write top value before calibration. Changing this while calibration is running results in bus fault.			

8.5.11 CMU_CLKEN0 - Clock Enable Register 0

Offset	Bit Position																															
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0			0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW	RW	RW	RW	RW
Name			BURTC	BURAM	PRS	GPIO	I2C2	ULFRCO	LFXO	LFRCO	FSRCO	HFXO0	HFRCOEM23	HFRCO0	DPLL0	SYSCFG	I2C1	I2C0	WDOG0	LETIMER0	AMUXCP0	ADC0			TIMER3	TIMER2	TIMER1	TIMER0	GPCRC0	LPWAES	LDMAXBAR0	LDMA0

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29	BURTC	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
28	BURAM	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
27	PRS	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
26	GPIO	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
25	I2C2	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
24	ULFRCO	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
23	LFXO	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
22	LFRCO	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
21	FSRCO	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
20	HFXO0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
19	HFRCOEM23	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
18	HFRCO0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
17	DPLL0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
16	SYSCFG	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
15	I2C1	0x0	RW	**Enable Bus Clock**

Bit	Name	Reset	Access	Description
	Enables module PCLK/HCLK			
14	I2C0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
13	WDOG0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
12	LETIMER0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
11	AMUXCP0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
10	ADC0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
9:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	TIMER3	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
6	TIMER2	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
5	TIMER1	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
4	TIMER0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
3	GPCRC0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
2	LPWAES	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
1	LDMAXBAR0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
0	LDMA0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			

8.5.12 CMU_CLKEN1 - Clock Enable Register 1

Offset	Bit Position																														
0x068	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x1	0x0			0x0	0x0	0x0	0x0		0x0	0x0	0x0		0x0	0x0		0x0		0x0				0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access	RW	RW	RW	RW	RW			RW	RW	RW	RW		RW	RW	RW		RW	RW		RW		RW				RW	RW	RW	RW	RW	RW	RW
Name	SYMCRYPTO	PIXELRZ1	PIXELRZ0	DEVINFO	DMEM			EUSART2	EUSART1	EUSART0	PCNT0		ACMP1	ACMP0	WDOG1		ICACHE0	SMU		L2ICACHE0		SEMAILBOXHOST				LEDDRV0	SEMAPHORE1	SEMAPHORE0	SOCPLL0	ETAMPDET	KSU	RPA

Bit	Name	Reset	Access	Description
31	SYMCRYPTO	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
30	PIXELRZ1	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
29	PIXELRZ0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
28	DEVINFO	0x1	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
27	DMEM	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
26:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	EUSART2	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
23	EUSART1	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
22	EUSART0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
21	PCNT0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19	ACMP1	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
18	ACMP0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
17	WDOG1	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15	ICACHE0	0x0	RW	**Enable Bus Clock**

Bit	Name	Reset	Access	Description
	Enables module PCLK/HCLK			
14	SMU	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	L2ICACHE0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	SEMAILBOXHOST	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
9:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	LEDDRV0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
5	SEMAPHORE1	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
4	SEMAPHORE0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
3	SOCPLL0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
2	ETAMPDET	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
1	KSU	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
0	RPA	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			

8.5.13 CMU_SYSCLKCTRL - System Clock Control

Offset	Bit Position																															
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0					0x0								0x1		
Access																	RW					RW								RW		
Name																	HCLKPRESC					PCLKPRESC								CLKSEL		

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:12	HCLKPRESC	0x0	RW	**HCLK Prescaler**
	Specifies the clock divider for HCLK			

Value	Mode	Description
0	DIV1	HCLK is SYSCLK divided by 1
1	DIV2	HCLK is SYSCLK divided by 2
3	DIV4	HCLK is SYSCLK divided by 4
7	DIV8	HCLK is SYSCLK divided by 8
15	DIV16	HCLK is SYSCLK divided by 16

Bit	Name	Reset	Access	Description
11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	PCLKPRESC	0x0	RW	**PCLK Prescaler**
	Specifies the clock divider for PCLK			

Value	Mode	Description
0	DIV1	PCLK is HCLK divided by 1
1	DIV2	PCLK is HCLK divided by 2

Bit	Name	Reset	Access	Description
9:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**
	Selects the clock source for SYSCLK.			

Value	Mode	Description
1	FSRCO	FSRCO is clocking SYSCLK
2	HFRCODPLL	HFRCODPLL is clocking SYSCLK
3	HFXO	HFXO is clocking SYSCLK
4	CLKIN0	CLKIN0 is clocking SYSCLK
5	SOCPLL	SOCPLL is clocking SYSCLK

8.5.14 CMU_TRACECLKCTRL - Debug Trace Clock Control

Offset	Bit Position																															
0x080	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0				0x1	
Access																											RW				RW	
Name																											PRESC				CLKSEL	

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	PRESC	0x0	RW	**TRACECLK Prescaler**

Clock prescaler for the TRACECLKIN of TPIU. Changing this while the TRCENA bit is set in the ARM M33 Debug Exception and Monitor Control Register (DEMCR) will result in a bus fault.

Value	Mode	Description
0	DIV1	TRACECLK is selected clock source divided by 1
1	DIV2	TRACECLK is selected clock source divided by 2
2	DIV3	TRACECLK is SYSCLK divided by 3
3	DIV4	TRACECLK is selected clock source divided by 4

Bit	Name	Reset	Access	Description
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

This bit is used to control which clock is for TRACE. Changing this while HOSTCPU TRACEENA is set results in bus fault.

Value	Mode	Description
0	DISABLE	TRACE clock is disable
1	SYSCLK	SYSCLK is driving TRACE
2	HFRCOEM23	HFRCOEM23 is driving TRACE
3	HFRCODPLLRT	HFRCODPLLRT is driving TRACE

8.5.15 CMU_EXPORTCLKCTRL - Export Clock Control

Offset	Bit Position																															
0x090	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0								0x0									0x0				0x0				0x0			
Access				RW								RW									RW				RW				RW			
Name				CLKOUTPRESC								PRESC									CLKOUTSEL2				CLKOUTSEL1				CLKOUTSEL0			

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28:24	CLKOUTPRESC	0x0	RW	**CLKOUT Prescaler**
	Specifies the clock divider for CLKOUT0/1/2 and CLKOUTHIDDEN.			
23:21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20:16	PRESC	0x0	RW	**EXPORTCLK Prescaler**
	Specifies the clock divider for EXPORTCLK (relative to SYSCLK).			
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	CLKOUTSEL2	0x0	RW	**Clock Output Select 2**
	Controls the clock output 2 multiplexer.			

Value	Mode	Description
0	DISABLED	CLKOUT2 is not clocked
1	HCLK	HCLK is clocking CLKOUT2
2	HFEXPCLK	EXPORTCLK is clocking CLKOUT2
3	ULFRCO	ULFRCO is clocking CLKOUT2
4	LFRCO	LFRCO is clocking CLKOUT2
5	LFXO	LFXO is clocking CLKOUT2
6	HFRCODPLL	HFRCODPLL is clocking CLKOUT2
7	HFXO	HFXO is clocking CLKOUT2
8	FSRCO	FSRCO is clocking CLKOUT2
9	HFRCOEM23	HFRCOEM23 is clocking CLKOUT2
10	SOCPLL	SOCPLL is clocking CLKOUT2

7:4	CLKOUTSEL1	0x0	RW	**Clock Output Select 1**
	Controls the clock output 1 multiplexer.			

Value	Mode	Description
0	DISABLED	CLKOUT1 is not clocked
1	HCLK	HCLK is clocking CLKOUT1
2	HFEXPCLK	EXPORTCLK is clocking CLKOUT1

Bit	Name	Reset	Access	Description
	3	ULFRCO		ULFRCO is clocking CLKOUT1
	4	LFRCO		LFRCO is clocking CLKOUT1
	5	LFXO		LFXO is clocking CLKOUT1
	6	HFRCODPLL		HFRCODPLL is clocking CLKOUT1
	7	HFXO		HFXO is clocking CLKOUT1
	8	FSRCO		FSRCO is clocking CLKOUT1
	9	HFRCOEM23		HFRCOEM23 is clocking CLKOUT1
	10	SOCPLL		SOCPLL is clocking CLKOUT1
3:0	CLKOUTSEL0	0x0	RW	**Clock Output Select 0**
	Controls the clock output 0 multiplexer.			

Value	Mode	Description
0	DISABLED	CLKOUT0 is not clocked
1	HCLK	HCLK is clocking CLKOUT0
2	HFEXPCLK	EXPORTCLK is clocking CLKOUT0
3	ULFRCO	ULFRCO is clocking CLKOUT0
4	LFRCO	LFRCO is clocking CLKOUT0
5	LFXO	LFXO is clocking CLKOUT0
6	HFRCODPLL	HFRCODPLL is clocking CLKOUT0
7	HFXO	HFXO is clocking CLKOUT0
8	FSRCO	FSRCO is clocking CLKOUT0
9	HFRCOEM23	HFRCOEM23 is clocking CLKOUT0
10	SOCPLL	SOCPLL is clocking CLKOUT0

8.5.16 CMU_DPLLREFCLKCTRL - Digital PLL Reference Clock Control

Offset	Bit Position																													
0x100	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x0	RW	**Clock Select**

Selects the clock source for DPLL reference. Changing this while DPLL is enabled results in bus fault.

Value	Mode	Description
0	DISABLED	DPLLREFCLK is not clocked
1	HFXO	HFXO is clocking DPLLREFCLK
2	LFXO	LFXO is clocking DPLLREFCLK
3	CLKIN0	CLKIN0 is clocking DPLLREFCLK

8.5.17 CMU_EM01GRPACLKCTRL - EM01 Peripheral Group A Clock Control

Offset	Bit Position																														
0x120	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x1		
Access																														RW		
Name																														CLKSEL		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for EM01 Group A Clock.

Value	Mode	Description
1	HFRCODPLL	HFRCODPLL is clocking EM01GRPACLK
2	HFXO	HFXO is clocking EM01GRPACLK
3	FSRCO	FSRCO is clocking EM01GRPACLK
4	HFRCOEM23	HFRCOEM23 is clocking EM01GRPACLK
5	HFRCODPLLRT	HFRCODPLL (retimed) is clocking EM01GRPACLK. Check with data sheet for frequency limitation when using retiming with voltage scaling.
6	HFXORT	HFXO (retimed) is clocking EM01GRPACLK. Check with data sheet for frequency limitation when using retiming with voltage scaling.

8.5.18 CMU_EM01GRPCCLKCTRL - EM01 Peripheral Group C Clock Control

Offset	Bit Position																														
0x128	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x1		
Access																														RW		
Name																														CLKSEL		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for EM01 Group C Clock.

Value	Mode	Description
1	HFRCODPLL	HFRCODPLL is clocking EM01GRPCCLK
2	HFXO	HFXO is clocking EM01GRPCCLK
3	FSRCO	FSRCO is clocking EM01GRPCCLK
4	HFRCOEM23	HFRCOEM23 is clocking EM01GRPCCLK
5	HFRCODPLLRT	HFRCODPLL (retimed) is clocking EM01GRPCCLK. Check with data sheet for frequency limitation when using retiming with voltage scaling.
6	HFXORT	HFXO (retimed) is clocking EM01GRPCCLK. Check with data sheet for frequency limitation when using retiming with voltage scaling.

8.5.19 CMU_EM23GRPACLKCTRL - EM23 Peripheral Group A Clock Control

Offset	Bit Position																													
0x140	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for EM23 Group A Clock.

Value	Mode	Description
1	LFRCO	LFRCO is clocking EM23GRPACLK
2	LFXO	LFXO is clocking EM23GRPACLK
3	ULFRCO	ULFRCO is clocking EM23GRPACLK

8.5.20 CMU_EM4GRPACLKCTRL - EM4 Peripheral Group A Clock Control

Offset	Bit Position																													
0x160	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for EM4 Group A Clock.

Value	Mode	Description
1	LFRCO	LFRCO is clocking EM4GRPACLK
2	LFXO	LFXO is clocking EM4GRPACLK
3	ULFRCO	ULFRCO is clocking EM4GRPACLK

8.5.21 CMU_ADCCLKCTRL - ADC Clock Control

Offset	Bit Position																													
0x180	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for for ADC. EM01GRPACLK should never be selected as the clock source when disabling the EM01GRPACLK (e.g. entering low energy modes).

Value	Mode	Description
1	EM01GRPACLK	EM01GRPACLK is clocking ADCCLK
2	FSRCO	FSRCO is clocking ADCCLK
3	HFRCOEM23	HFRCOEM23 is clocking ADCCLK

8.5.22 CMU_WDOG0CLKCTRL - WDOG0 Clock Control

Offset	Bit Position																													
0x200	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x1		
Access																														RW		
Name																														CLKSEL		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for WDOG0.

Value	Mode	Description
1	LFRCO	LFRCO is clocking WDOG0CLK
2	LFXO	LFXO is clocking WDOG0CLK
3	ULFRCO	ULFRCO is clocking WDOG0CLK
4	HCLKDIV1024	HCLKDIV1024 is clocking WDOG0CLK

8.5.23 CMU_EUSART0CLKCTRL - EUSART0 Clock Control

Offset	Bit Position																														
0x220	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x1		
Access																														RW		
Name																														CLKSEL		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

This bit controls which clock is used for EUSART0. EM01GRPCCLK should never be selected as the clock source when disabling the EM01GRPCCLK (e.g. entering low energy modes).

Value	Mode	Description
0	DISABLED	EUSART0 is not clocked
1	EM01GRPCCLK	EM01GRPCCLK is clocking EUSART0
2	HFRCOEM23	HFRCOEM23 is clocking EUSART0
3	LFRCO	LFRCO is clocking EUSART0
4	LFXO	LFXO is clocking EUSART0

8.5.24 CMU_SYSRTC0CLKCTRL - SYSRTC0 Clock Control

Offset	Bit Position																														
0x240	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for SYSRTC0.

Value	Mode	Description
1	LFRCO	LFRCO is clocking SYSRTC0CLK
2	LFXO	LFXO is clocking SYSRTC0CLK
3	ULFRCO	ULFRCO is clocking SYSRTC0CLK

8.5.25 CMU_PCNT0CLKCTRL - PCNT0 Clock Control

Offset	Bit Position																													
0x270	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**
	This bit controls which clock is used for PCNT0.			

Value	Mode	Description
0	DISABLED	PCNT0 is not clocked
1	EM23GRPACLK	EM23GRPACLK is clocking PCNT0
2	PCNTS0	External pin PCNT_S0 is clocking PCNT0

8.5.26 CMU_RADIOCLKCTRL - Radio Clock Control

Offset	Bit Position																													
0x280	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															0x0
Access	RW																															RW
Name	DBGCLK																															EN

Bit	Name	Reset	Access	Description
31	DBGCLK	0x0	RW	**Enable Clock for Debugger**
	When set to 1, this forces radio busmatrix and RAC clocks to run, allowing RAC sequencer debugger to stay attached.			
30:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	EN	0x0	RW	**Enable**
	Enables radio bridge clock			

8.5.27 CMU_HFXO0LFCLKCTRL - HFXO0 LF Clock Control

Offset	Bit Position																													
0x29C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	
Access																															RW	
Name																															CLKSEL	

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	CLKSEL	0x1	RW	**Clock Select**

This bit controls which low frequency clock is used for HFXO startup time measurement

Value	Mode	Description
0	DISABLED	HFXO0 LF clock is not generated
1	LFXO	LFXO is clocking for startup time measurement
2	LFRCO	LFRCO is clocking for startup time measurement

8.5.28 CMU_EM01GRPDCLKCTRL - EM01 Peripheral Group D Clock Control

Offset	Bit Position																													
0x2A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x1		
Access																														RW		
Name																														CLKSEL		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for EM01 Group D Clock.

Value	Mode	Description
1	HFRCODPLL	HFRCODPLL is clocking EM01GRPDCLK
2	HFXO	HFXO is clocking EM01GRPDCLK
3	FSRCO	FSRCO is clocking EM01GRPDCLK
4	HFRCOEM23	HFRCOEM23 is clocking EM01GRPDCLK

8.5.29 CMU_I2C0CLKCTRL - I2C0 Clock Control

Offset	Bit Position																														
0x2AC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x1		
Access																														RW		
Name																														CLKSEL		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	CLKSEL	0x1	RW	**Clock Select**

This bit controls which clock is used for I2C0. EM01GRPCCLK should never be selected as the clock source when disabling the EM01GRPCCLK (e.g. entering low energy modes).

Value	Mode	Description
0	DISABLED	I2C0 is not clocked
1	EM01GRPDCLK	EM01GRPCCLK is clocking I2C0
2	HFRCOEM23	HFRCOEM23 is clocking I2C0
3	LFRCO	LFRCO is clocking I2C0
4	LFXO	LFXO is clocking I2C0

8.5.30 CMU_PIXELRZCLKCTRL - PIXELRZ Clock Control

Offset	Bit Position																															
0x2B0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																						0x1					0x9					
Access																						RW					RW					
Name																						CLKSEL					PRESC					

Bit	Name	Reset	Access	Description
31:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:8	CLKSEL	0x1	RW	**Clock Select**

Selects the clock source for PIXELRZ0/1 Clock.

Value	Mode	Description
1	HFRCODPLL	HFRCODPLL is clocking PIXELRZ0/1
2	HFXO	HFXO is clocking PIXELRZ0/1
3	FSRCO	FSRCO is clocking PIXELRZ0/1
4	HFRCOEM23	HFRCOEM23 is clocking PIXELRZ0/1

7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:0	PRESC	0x9	RW	**PIXELRZ Clock Prescaler**

Clock prescaler for PIXELRZ0/1. Change to ensure PIXELRZ clock is at or below maximum allowed frequency.

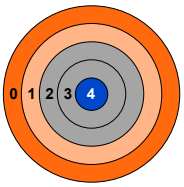
Value	Mode	Description
0	DIV1	PIXELRZ clock divided by 1
1	DIV2	PIXELRZ clock divided by 2
2	DIV3	PIXELRZ clock divided by 3
3	DIV4	PIXELRZ clock divided by 4
9	DIV10	PIXELRZ clock divided by 10
15	DIV16	PIXELRZ clock divided by 16

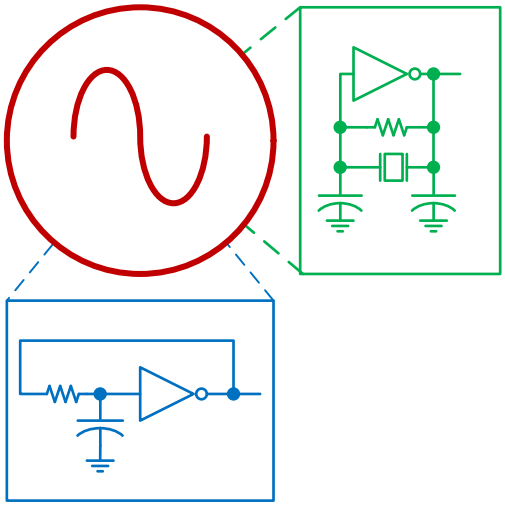
8.5.31 CMU_CLKENHV - Clock Enable Register 2

Offset	Bit Position																													
0x2B4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0	0x0	0x0			0x0		0x0
Access																									RW	RW	RW			RW		RW
Name																									SEPORTAL	LPW0PORTAL	HOSTPORTAL			SCRATCHPAD		SYSRTC0

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	SEPORTAL	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
6	LPW0PORTAL	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
5	HOSTPORTAL	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
4:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	SCRATCHPAD	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	SYSRTC0	0x0	RW	**Enable Bus Clock**
	Enables module PCLK/HCLK			

9. Oscillators





Quick Facts

What?

The SiXG301 has a wide range of high-frequency and low-frequency oscillators.

Why?

The high-frequency oscillators support high-speed operation. The low-frequency oscillators provide low-frequency clocks for the low-energy peripherals and accurate timebase for RTC functions.

How?

The HFXO supports high-frequency crystal oscillators. The LFXO supports 32.768 kHz crystal oscillators. The RC oscillators are internal oscillators that require no external components.

9.1 Introduction

The SiXG301 has several oscillators. This chapter contains a detailed function description and register descriptions for each oscillator. The CMU chapter includes information on how to select clock sources. Each oscillator may require some initial configuration or calibration before being enabled. The CMU supports clock on demand and can enable and disable oscillators. Therefore, it is important to properly configure each oscillator before selecting it as a clock source in the CMU.

9.2 HFXO - High Frequency Crystal Oscillator

9.2.1 Introduction

The High Frequency Crystal Oscillator (HFXO) uses an external high frequency crystal and provides a sequencer for starting up the crystal safely and reliably, while minimize energy consumption. An external sine wave clock source can also be used in the absence of a crystal.

9.2.2 Features

- Optimized for 38.4 MHz crystals
- Multiple programming options of start-up parameters to enable optimization of different crystals, supporting a wide range of ESR and ESL
- External sine wave input qualified with amplitude detection
- Programmable two-phase start-up to minimize energy consumption
- Built-in current optimization (Automatic oscillation amplitude control)
- Programmable clock injection for faster startup time
- Independent on-chip frequency tuning capacitors
- Hardware request for on-demand enable/disable
- Register lock
- Hardware startup time measurement

9.2.3 Functional Description

9.2.3.1 Enabling and Disabling

While the HFXO supports on-demand clocking, it is generally recommended to manually manage the HFXO, at least initially, because it requires software configuration and has a long start-up time. Software can set the FORCEEN to start HFXO and keep it enabled even if it is not selected as a clock source.

However, once started and before sleep mode entry, switching the HFXO to on-demand mode may be desirable. This allows the MCU to enter a low energy mode and then restart the HFXO automatically upon wake. (During EM1P the HFXO can be conditionally started, depending on the wake-up trigger source.)

The HFXO can be enabled and disabled via both hardware and software mechanisms. Enabling via software is done by setting the FORCEEN bit in the HFXO_CTRL register. Disabling via software is done by setting the DISONDEMAND bit and clearing FORCEEN bit in the HFXO_CTRL register. The hardware controlled on-demand mode is enabled by clearing the FORCEEN and DISONDEMAND bits in the HFXO_CTRL register. Once configured the on-demand mode hardware can autonomously start and stop the HFXO based on various peripheral clock requests in combination with clock switch selections in the CMU. The HFXO is automatically stopped when entering EM4. Hardware can also stop the HFXO via hardware in response to change in peripheral requests and clock switch selections in the CMU.

9.2.3.2 Start-up Time

The start-up time differs for different crystals and the HFXO has a configurable time-out to accommodate each crystal type. Software configures the timeout by setting the various TIMEOUT bit fields of the HFXO_XTALCFG register. The time-out delays the assertion of the RDY signal for HFXO. The programmed timeout should allow enough time for the oscillator to stabilize. The time-out can be optimized for the chosen crystal used in the application.

The start-up behavior of the HFXO also depends on how and how long the HFXO is disabled.

9.2.3.3 Configuration

The High Frequency Crystal Oscillator needs to be configured to ensure safe start-up for the given crystal. Refer to the device data sheet and application notes for guidelines in selecting correct components and crystals as well as for configuration trade-offs.

The HFXO crystal is connected to the HFXTAL_I/HFXTAL_O pins as shown in Figure 9.1. HFXO Pin Connection.

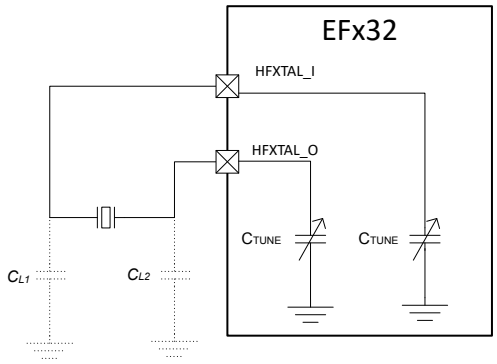


Figure 9.1. HFXO Pin Connection

Upon enabling the HFXO, a hardware state machine sequentially applies the configurable start-up state, intermediate start-up state, and steady state control settings from the HFXO_XTALCFG and HFXO_XTALCTRL registers. After reaching the steady operation state of the HFXO, it is recommended to further optimize current consumption using the Core Bias Optimization Algorithm to trade off noise and current consumption.

Write the configuration values, which depends on the crystal's CL, RESR and oscillation frequency, into HFXO_XTALCFG and HFXO_XTALCTRL registers.

- COREBIASSTARTUP (HFXO_XTALCFG) - current setting applied at start-up time
- COREBIASSTARTUPI (HFXO_XTALCFG) - current setting applied at intermediate start-up time
- COREBIASANA (HFXO_XTALCTRL) - current setting applied at steady state
- CTUNEXISTARTUP (HFXO_XTALCFG) - tuning cap setting for XI applied at start-up time
- CTUNEXIANA (HFXO_XTALCTRL) - tuning cap setting for XI applied at steady state
- CTUNEXOSTARTUP (HFXO_XTALCFG) - tuning cap setting for XO applied at start-up time
- CTUNEXOANA (HFXO_XTALCTRL) - tuning cap setting for XO applied at steady state
- CTUNEFIXANA (HFXO_XTALCTRL) - fixed tuning cap setting applied throughout
- TIMEOUTSTEADY (HFXO_XTALCFG) - duration for the steady state settling time
- TIMEOUTCBLSB (HFXO_XTALCFG) - duration for the optimization settling after each step

All HFXO configuration needs to be performed prior to enabling the HFXO, whether via software by setting FORCEEN bit field, or allowing hardware request by clearing DISONDEMAND bit field in the HFXO_CTRL register.

By default, the HFXO is started in crystal mode, but it is possible to connect an active external sine or clipped sine wave clock source to the HFXTAL_I pin of the HFXO. By configuring the MODE field in HFXO_CFG to EXTCLK, the HFXO can be bypassed and the source clock can be provided through the HFXTAL_I pin.

9.2.3.4 Status Flags

The ENS flag in the HFXO_STATUS indicates if the HFXO has been successfully enabled. Once the HFXO oscillation amplitude has exceeded the start-up threshold and intermediate start-up threshold, the steady state settling timeout begins. When the steady state timeout has expired, the HFXO is ready for use as indicated by the RDY flag in the HFXO_STATUS. Once Core Bias Optimization is enabled, the COREBIASOPTRDY flag in the CMU_STATUS register indicates when the optimization is ready. It is advised to wait for this flag before using the HFXO, because optimization can cause minor disturbance to the oscillator frequency.

9.2.3.5 On-Demand Clocking

Hardware can request to enable the HFXO by setting the HFXO_STATUS.HWREQ bit field. The HFXO can also optionally be configured via the HFXO_CTRL.DISONDEMAND to shut down when no hardware request is present. This is known as on-demand clocking and allows the oscillator to be controlled without any software intervention. On-demand HFXO enable can be used, for example, upon wake-up of the Radio Controller (RAC). The RAC module always requires the HFXO for its operation. Any hardware request for HFXO, including request from RAC, is indicated in the HWREQ bit field of the HFXO_STATUS register. This request enables the HFXO, provided that DISONDEMAND bit field is cleared in HFXO_CTRL register. The HFXO is only disabled by hardware upon entry to an unsupported energy mode.

The HFXO analog circuitry can optionally continue operating with the clock output shut off when the HFXO is disabled. This is configured by setting the KEEPWARM bit in HFXO_STATUS.

9.2.3.6 Interrupts

RDYIF and COREBIASOPTRDYIF are interrupt flags as well as status flags. This allows software flexibility to implement interrupt service routine or polling loop for these events. When steady state timeout has exceeded, sticky RDYIF is set until it is cleared by software. If optimization is enabled, sticky COREBIASOPTRDYIF is set when optimization is completed successfully. However, if optimization fails to complete, sticky COREBIASOPTERRIF is set, and the HFXO control state machine stays in the error state until the oscillator is disabled. Similarly, if HFXO fails to start-up, meaning it has not reached the steady state, sticky DNSERRIF is set. The HFXO control state machine stays in the error state until the oscillator is disabled.

9.2.3.7 Protection

It is possible to lock the control registers, configuration registers, and command register to prevent unintended software writes to critical clock settings. This is controlled by the HFXO_LOCK register. A LOCK bit is available in HFXO_STATUS register. Furthermore, these registers are locked automatically by hardware to prevent clock domain crossing malfunction. To gain access to these registers while oscillator is in steady operation state, set FORCEEN to 1, then set DISONDEMAND to 1 in the HFXO_CTRL register.

A FSMLOCK bit in HFXO_STATUS register indicates when it is safe for software to update control registers and configuration registers. When software is finished with updates, put the oscillator back to on-demand mode by clearing DISONDEMAND to 0, followed by clearing FORCEEN to 0 in the HFXO_CTRL register. While DISONDEMAND is 0, FSMLOCK is always set, even if hardware is not requesting. This is to prevent a race condition between software access and hardware lock.

9.2.3.8 Tuning

While the oscillator is running in steady operation state, it may be desirable to change control settings. One example is frequency tuning by modifying the tuning capacitance via CTUNEXIANA and CTUNEXOANA fields in the HFXO_XTALCTRL register. When tuning, care should be taken to make small changes to the CTUNE registers. Ideally, change the CTUNE registers by one LSB at a time and alternate between the XI and XO registers. Sufficient wait time for settling, on the order of TIMEOUTSTEADY, should pass before new frequency measurement is taken.

Note: While the HFXO can support crystals with a tuning range of 38 MHz to 40 MHz, the radio specifically requires a 38.4 MHz crystal. There may also be specific crystal tolerance requirements for each RF protocol supported by the radio.

9.2.4 HFXO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	HFXO_IPVERSION	R	IP Version ID
0x010	HFXO_XTALCFG	RW SYNC	Crystal Configuration Register
0x018	HFXO_XTALCTRL	RWH SYNC	Crystal Control Register
0x020	HFXO_CFG	RW SYNC	Configuration Register
0x028	HFXO_SLEEPYXTALCFG0	RW SYNC	Sleepy Crystal Configuration 0
0x02C	HFXO_SLEEPYXTALCFG1	RW SYNC	Sleepy Crystal Configuration 1
0x030	HFXO_CTRL	RWH SYNC	Control Register
0x040	HFXO_EXTCLKPKDETCTRL	RW SYNC	Peak Detector Control Register
0x054	HFXO_CMD	W SYNC	Command Register
0x058	HFXO_STATUS	RH	Status Register
0x05C	HFXO_AVGSTARTUPTIME	RH	Average Startup Time Register
0x070	HFXO_IF	RWH INTFLAG	Interrupt Flag Register
0x074	HFXO_IEN	RW	Interrupt Enable Register
0x080	HFXO_LOCK	W	Configuration Lock Register
0x1000	HFXO_IPVERSION_SET	R	IP Version ID
0x1010	HFXO_XTALCFG_SET	RW SYNC	Crystal Configuration Register
0x1018	HFXO_XTALCTRL_SET	RWH SYNC	Crystal Control Register
0x1020	HFXO_CFG_SET	RW SYNC	Configuration Register
0x1028	HFXO_SLEEPYXTALCFG0_SET	RW SYNC	Sleepy Crystal Configuration 0
0x102C	HFXO_SLEEPYXTALCFG1_SET	RW SYNC	Sleepy Crystal Configuration 1
0x1030	HFXO_CTRL_SET	RWH SYNC	Control Register
0x1040	HFXO_EX-TCLKPKDETCTRL_SET	RW SYNC	Peak Detector Control Register
0x1054	HFXO_CMD_SET	W SYNC	Command Register
0x1058	HFXO_STATUS_SET	RH	Status Register
0x105C	HFXO_AVGSTARTUPTIME_SET	RH	Average Startup Time Register
0x1070	HFXO_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1074	HFXO_IEN_SET	RW	Interrupt Enable Register
0x1080	HFXO_LOCK_SET	W	Configuration Lock Register
0x2000	HFXO_IPVERSION_CLR	R	IP Version ID
0x2010	HFXO_XTALCFG_CLR	RW SYNC	Crystal Configuration Register
0x2018	HFXO_XTALCTRL_CLR	RWH SYNC	Crystal Control Register
0x2020	HFXO_CFG_CLR	RW SYNC	Configuration Register
0x2028	HFXO_SLEEPYXTALCFG0_CLR	RW SYNC	Sleepy Crystal Configuration 0
0x202C	HFXO_SLEEPYXTALCFG1_CLR	RW SYNC	Sleepy Crystal Configuration 1
0x2030	HFXO_CTRL_CLR	RWH SYNC	Control Register
0x2040	HFXO_EX-TCLKPKDETCTRL_CLR	RW SYNC	Peak Detector Control Register
0x2054	HFXO_CMD_CLR	W SYNC	Command Register
0x2058	HFXO_STATUS_CLR	RH	Status Register

Offset	Name	Type	Description
0x205C	HFXO_AVGSTARTUPTIME_CLR	RH	Average Startup Time Register
0x2070	HFXO_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2074	HFXO_IEN_CLR	RW	Interrupt Enable Register
0x2080	HFXO_LOCK_CLR	W	Configuration Lock Register
0x3000	HFXO_IPVERSION_TGL	R	IP Version ID
0x3010	HFXO_XTALCFG_TGL	RW SYNC	Crystal Configuration Register
0x3018	HFXO_XTALCTRL_TGL	RWH SYNC	Crystal Control Register
0x3020	HFXO_CFG_TGL	RW SYNC	Configuration Register
0x3028	HFXO_SLEEPYXTALCFG0_TGL	RW SYNC	Sleepy Crystal Configuration 0
0x302C	HFXO_SLEEPYXTALCFG1_TGL	RW SYNC	Sleepy Crystal Configuration 1
0x3030	HFXO_CTRL_TGL	RWH SYNC	Control Register
0x3040	HFXO_EX-TCLKPKDETCTRL_TGL	RW SYNC	Peak Detector Control Register
0x3054	HFXO_CMD_TGL	W SYNC	Command Register
0x3058	HFXO_STATUS_TGL	RH	Status Register
0x305C	HFXO_AVGSTARTUPTIME_TGL	RH	Average Startup Time Register
0x3070	HFXO_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3074	HFXO_IEN_TGL	RW	Interrupt Enable Register
0x3080	HFXO_LOCK_TGL	W	Configuration Lock Register

9.2.5 HFXO Register Description

9.2.5.1 HFXO_IPVERSION - IP Version ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x5																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x5	R	**IP Version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

9.2.5.2 HFXO_XTALCFG - Crystal Configuration Register

Offset	Bit Position																															
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x2				0xB				0xC					0x0				0x0					0x8						0x8			
Access	RW				RW				RW					RW				RW					RW						RW			
Name	TINJ				TIMEOUTCBLSB				TIMEOUTSTEADY					CTUNEXOSTARTUP				CTUNEXISTARTUP					COREBIASSTARTUP						COREBIASSTARTUPI			

Bit	Name	Reset	Access	Description
31:28	TINJ	0x2	RW	**Injection Time Configuration**

Injection Time Configuration with ~20MHz (divide by 2 in analog) counter clock period

Value	Mode	Description
0	T10CYCLES	Clock injection for 10 xtal cycles (0.5us)
1	T20CYCLES	Clock injection for 20 xtal cycles (1us)
2	T30CYCLES	Clock injection for 30 xtal cycles (1.5us)
3	T40CYCLES	Clock injection for 40 xtal cycles (2us)
4	T50CYCLES	Clock injection for 50 xtal cycles (2.5us)
5	T60CYCLES	Clock injection for 60 xtal cycles (3us)
6	T70CYCLES	Clock injection for 70 xtal cycles (3.5us)
7	T80CYCLES	Clock injection for 80 xtal cycles (4us)
8	T90CYCLES	Clock injection for 90 xtal cycles (4.5us)
9	T100CYCLES	Clock injection for 100 xtal cycles (5us)
10	T110CYCLES	Clock injection for 110 xtal cycles (5.5us)
11	T120CYCLES	Clock injection for 120 xtal cycles (6us)
12	T130CYCLES	Clock injection for 130 xtal cycles (6.5us)
13	T140CYCLES	Clock injection for 140 xtal cycles (7us)
14	T180CYCLES	Clock injection for 180 xtal cycles (9us)
15	T210CYCLES	Clock injection for 210 xtal cycles (10.5us)

Bit	Name	Reset	Access	Description
27:24	TIMEOUTCBLSB	0xB	RW	**Core Bias LSB Change Timeout**

wait duration for the COREBIAS change to settle out, used at each step of COREBIAS optimization algorithm

Value	Mode	Description
0	T8US	The core bias LSB change timeout is set to 8 us minimum. The maximum can be +40%.
1	T20US	The core bias LSB change timeout is set to 20 us minimum. The maximum can be +40%.
2	T41US	The core bias LSB change timeout is set to 41 us minimum. The maximum can be +40%.

Bit	Name	Reset	Access	Description
	3	T62US		The core bias LSB change timeout is set to 62 us minimum. The maximum can be +40%.
	4	T83US		The core bias LSB change timeout is set to 83 us minimum. The maximum can be +40%.
	5	T104US		The core bias LSB change timeout is set to 104 us minimum. The maximum can be +40%.
	6	T125US		The core bias LSB change timeout is set to 125 us minimum. The maximum can be +40%.
	7	T166US		The core bias LSB change timeout is set to 166 us minimum. The maximum can be +40%.
	8	T208US		The core bias LSB change timeout is set to 208 us minimum. The maximum can be +40%.
	9	T250US		The core bias LSB change timeout is set to 250 us minimum. The maximum can be +40%.
	10	T333US		The core bias LSB change timeout is set to 333 us minimum. The maximum can be +40%.
	11	T416US		The core bias LSB change timeout is set to 416 us minimum. The maximum can be +40%.
	12	T833US		The core bias LSB change timeout is set to 833 us minimum. The maximum can be +40%.
	13	T1250US		The core bias LSB change timeout is set to 1250 us minimum. The maximum can be +40%.
	14	T2083US		The core bias LSB change timeout is set to 2083 us minimum. The maximum can be +40%.
	15	T3750US		The core bias LSB change timeout is set to 3750 us minimum. The maximum can be +40%.
23:20	TIMEOUTSTEADY	0xC	RW	**Steady State Timeout**
	wait duration for the steady state settings to settle out			

Value	Mode	Description
0	T4US	The steady state timeout is set to 4 us minimum. The maximum can be +40%.
1	T16US	The steady state timeout is set to 16 us minimum. The maximum can be +40%.
2	T41US	The steady state timeout is set to 41 us minimum. The maximum can be +40%.
3	T83US	The steady state timeout is set to 83 us minimum. The maximum can be +40%.
4	T125US	The steady state timeout is set to 125 us minimum. The maximum can be +40%.
5	T166US	The steady state timeout is set to 166 us minimum. The maximum can be +40%.
6	T208US	The steady state timeout is set to 208 us minimum. The maximum can be +40%.
7	T250US	The steady state timeout is set to 250 us minimum. The maximum can be +40%.
8	T333US	The steady state timeout is set to 333 us minimum. The maximum can be +40%.
9	T416US	The steady state timeout is set to 416 us minimum. The maximum can be +40%.

Bit	Name	Reset	Access	Description
	10	T500US		The steady state timeout is set to 500 us minimum. The maximum can be +40%.
	11	T666US		The steady state timeout is set to 666 us minimum. The maximum can be +40%.
	12	T833US		The steady state timeout is set to 833 us minimum. The maximum can be +40%.
	13	T1666US		The steady state timeout is set to 1666 us minimum. The maximum can be +40%.
	14	T2500US		The steady state timeout is set to 2500 us minimum. The maximum can be +40%.
	15	T4166US		The steady state timeout is set to 4166 us minimum. The maximum can be +40%.
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18:16	CTUNEXOSTARTUP	0x0	RW	**Startup Tuning Capacitance on XO**
	4 most significant bits of CTUNEXOANA applied during startup phase			
15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:12	CTUNEXISTARTUP	0x0	RW	**Startup Tuning Capacitance on XI**
	4 most significant bits of CTUNEXIANA applied during startup phase			
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:6	COREBIASSTARTUP	0x8	RW	**Startup Core Bias Current**
	6 most significant bits of COREBIASANA applied during startup phase			
5:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	COREBIASSTARTUPI	0x8	RW	**Intermediate Startup Core Bias Current**
	6 most significant bits of COREBIASANA applied during intermediate startup phase			

9.2.5.3 HFXO_XTALCTRL - Crystal Control Register

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0					0x1		0x20								0x20								0x78								
Access	RW					RW		RW								RW								RW								
Name	SKIPCOREBIASOPT					CTUNEFIXANA		CTUNEXOANA								CTUNEXIANA								COREBIASANA								

Bit	Name	Reset	Access	Description
31	SKIPCOREBIASOPT	0x0	RW	**Skip Core Bias Optimization**
	Set to skip the core bias current optimization algorithm at next startup. Reuse the value stored in COREBIASANA. At the successful completion of core bias current optimization algorithm, hardware sets this bit to skip optimization during subsequent startup.			
30:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:25	CTUNEFIXANA	0x1	RW	**Fixed Tuning Capacitance**
	Adds or removes fixed capacitance on XI or XO			

Value	Mode	Description
0	NONE	Remove fixed capacitance on XO node
1	SEL0	Adds fixed capacitance of 1.31pF on XO node
2	SEL1	Adds fixed capacitance of 2.62pF on XO node
3	SEL2	Adds fixed capacitance of 3.93pF on XO node

24:17	CTUNEXOANA	0x20	RW	**Tuning Capacitance on XO**
	Approximately 80fF per step. 0 is min. 255 is max.			
16:9	CTUNEXIANA	0x20	RW	**Tuning Capacitance on XI**
	Approximately 80fF per step. 0 is min. 255 is max.			
8:0	COREBIASANA	0x78	RW	**Core Bias Current**
	Approximately 5uA per step			

9.2.5.4 HFXO_CFG - Configuration Register

Offset	Bit Position																														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0		0x1	0x1																0x0		0x0	0x0						0x0	0x0	
Access		RW		RW	RW																RW		RW	RW						RW	RW	
Name		SQBUFENSTARTUPI		FORCELFTIMEOUTPRS	FORCELFTIMEOUTSYSRTC																NUMSTUPMEAS		STUPMEASEN	SLEEPYXTALSUPEN						ENXIDCBIASANA	MODE	

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30	SQBUFENSTARTUPI	0x0	RW	**Early enable for Squaring Buffer**
	Early enable for Squaring Buffer to enable in startup intermediate phase			
29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	FORCELFTI-MEOUTPRS	0x1	RW	**Force PRS Low Frequency Timeout**
	For deterministic timeout, clear this bit and configure PRS to trigger based on 32kHz timer like RTCC.			
27	FORCELFTIMEOUT-SYSRTC	0x1	RW	**Force SYSRTC Low Frequency Timeout**
	For deterministic timeout, clear this bit and configure SYSRTC to trigger based on 32kHz clock source.			
26:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:10	NUMSTUPMEAS	0x0	RW	**Number of startup time measurements**
	Configure number of startup time measurement samples to be captured and averaged over			

Value	Mode	Description
0	MEASURE1	Measure and average over 1 startup time samples
1	MEASURE8	Measure and average over 8 startup time samples
2	MEASURE16	Measure and average over 16 startup time samples

9	STUPMEASEN	0x0	RW	**Enable startup time measurement logic**
	Enable startup time measurement logic and averaging logic			
8	SLEEPYXTALSUPEN	0x0	RW	**Enable sleepy crystal startup logic**
	Enable sleepy crystal startup logic			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	ENXIDCBIASANA	0x0	RW	**Enable XI Internal DC Bias**
	Set to enable internal DC bias. Bit is ignored in XTAL mode.			
1:0	MODE	0x0	RW	**Crystal Oscillator Mode**

Bit	Name	Reset	Access	Description
	Set this to configure the external source for the HFXO.			

Value	Mode	Description
0	XTAL	crystal oscillator
1	EXTCLK	external sinusoidal clock can be supplied on XI pin.
2	EXTCLKPKDET	external sinusoidal clock can be supplied on XI pin (peak detector used).

9.2.5.5 HFXO_SLEEPYXTALCFG0 - Sleepy Crystal Configuration 0

Offset	Bit Position																															
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x2				0x0				0x2				0x0							0x5				0x1				0x8			
Access		RW				RW				RW				RW							RW				RW				RW			
Name		CTUNEXOANASLEEPY				CTUNEXOSUPSLEEPY				CTUNEXIANASLEEPY				CTUNEXISUPSLEEPY							PKDETTHANASLEEPY				PKDETTHSUPISLEEPY				PKDETTHSUPSLEEPY			

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30:28	CTUNEXOANASLEEPY	0x2	RW	**XI pin tuning cap after startup**
	XO pin tuning cap during startup intermediate and steady states			
27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:24	CTUNEXOSUPSLEEPY	0x0	RW	**XI pin tuning cap during startup**
	XO pin tuning cap during startup state			
23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	CTUNEXIANASLEEPY	0x2	RW	**XI pin tuning cap after startup**
	XI pin tuning cap during startup intermediate and steady states			
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18:16	CTUNEXISUPSLEEPY	0x0	RW	**XI pin tuning cap during startup**
	XI pin tuning cap during startup state			
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	PKDETTHANASLEEPY	0x5	RW	**Steady State Peak Detector Threshold**
	Steady State Peak Detector Threshold			
7:4	PKDETTHSUPISLEEPY	0x1	RW	**Startup Intermediate Pkdet Threshold**
	Startup Intermediate Peak Detector Threshold			
3:0	PKDETTHSUPSLEEPY	0x8	RW	**Startup Peak Detector Threshold**
	Startup Peak Detector Threshold			

9.2.5.6 HFXO_SLEEPYXTALCFG1 - Sleepy Crystal Configuration 1

Offset	Bit Position																														
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x1FF									0x1FF									0x1FF								
Access						RW									RW									RW								
Name						COREBIASANASLEEPY									COREBIASSUPISLEEPY									COREBIASSUPSLEEPY								

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:18	COREBIASANASLEEPY	0x1FF	RW	**Steady Phase Core Bias Setting**
	Core Bias Setting During Steady Phase			
17:9	COREBIASSUPISLEEPY	0x1FF	RW	**Startup Intermediate Phase Core Bias**
	Core Bias Setting During Startup Intermediate Phase			
8:0	COREBIASSUPSLEEPY	0x1FF	RW	**Startup Phase Core Bias**
	Core Bias Setting During Startup Phase			

9.2.5.7 HFXO_CTRL - Control Register

Offset	Bit Position																															
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x1		0x1	0x1					0x0		0x0	0x0	0x0				0x0					0x1	0x0	0x0		0x0		
Access					RW		RW	RW					RW		RW	RW	RW				RW					RW	RW	RW		RW		
Name					DISONDEMANDSYSRTC		DISONDEMANDPRS	DISONDEMAND					FORCEENSYSRTC		FORCEENPRS	FORCEEN	PRSSTATUSSEL1				PRSSTATUSSEL0					FORCECTUNEMAX	FORCEXO2GNDANA	FORCEXI2GNDANA		KEEPWARM		

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	DISONDEMAND-SYSRTC	0x1	RW	**Disable On-demand For SYSRTC**
	Disable On Demand for the SYSRTC request interface.			
26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	DISONDEMANDPRS	0x1	RW	**Disable On-demand For PRS**
	Disable On Demand for the PRS request interface.			
24	DISONDEMAND	0x1	RW	**Disable On-demand For Digital Clock**
	Disable On Demand for the digital clock request interface.			
23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19	FORCEENSYSRTC	0x0	RW	**Force SYSRTC Oscillator Request**
	Force SYSRTC oscillator request.			
18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	FORCEENPRS	0x0	RW	**Force PRS Oscillator Request**
	Force PRS oscillator request.			
16	FORCEEN	0x0	RW	**Force Digital Clock Request**
	Force digital clock request.			
15:12	PRSSTATUSSEL1	0x0	RW	**PRS Status 1 Output Select**
	Mux select for various status signals to be output through PRS.			

Value	Mode	Description
0	DISABLED	PRS mux outputs 0
1	ENS	PRS mux outputs enabled status
2	COREBIASOPTRDY	PRS mux outputs core bias optimization ready status
3	RDY	PRS mux outputs ready status
4	PRSRDY	PRS mux outputs PRS ready status
5	SYSRTCRDY	PRS mux outputs SYSRTC ready status

Bit	Name	Reset	Access	Description
	9	HWREQ		PRS mux outputs oscillator requested by digital clock status
	10	PRSHWREQ		PRS mux outputs oscillator requested by PRS request status
	11	SYSRTCHWREQ		PRS mux outputs oscillator requested by SYSRTC request status
11:8	PRSSTATUSSEL0	0x0	RW	**PRS Status 0 Output Select**
	Mux select for various status signals to be output through PRS.			
	Value	Mode		Description
	0	DISABLED		PRS mux outputs 0
	1	ENS		PRS mux outputs enabled status
	2	COREBIASOPTRDY		PRS mux outputs core bias optimization ready status
	3	RDY		PRS mux outputs ready status
	4	PRSRDY		PRS mux outputs PRS ready status
	5	SYSRTCRDY		PRS mux outputs SYSRTC ready status
	9	HWREQ		PRS mux outputs oscillator requested by digital clock status
	10	PRSHWREQ		PRS mux outputs oscillator requested by PRS request status
	11	SYSRTCHWREQ		PRS mux outputs oscillator requested by SYSRTC request status
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	FORCECTUNEMAX	0x1	RW	**Force Tuning Cap to Max Value**
	When oscillator is disabled, force tuning capacitor to maximum value. Set this bit to 1 in XTAL mode to prevent overshoot upon disable.			
5	FORCEXO2GNDANA	0x0	RW	**Force XO Pin to Ground**
	Set to enable grounding of XO pin.			
	Value	Mode		Description
	0	DISABLE		Disabled (not pulled)
	1	ENABLE		Enabled (pulled)
4	FORCEXI2GNDANA	0x0	RW	**Force XI Pin to Ground**
	Set to enable grounding of XI pin. Do not enable if MODE=EXTCLK and an external source is supplied.			
	Value	Mode		Description
	0	DISABLE		Disabled (not pulled)
	1	ENABLE		Enabled (pulled)
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	KEEPWARM	0x0	RW	**Keep Warm**
	Upon disable, if this bit is set, analog oscillator will keep running, while clock output is shutoff. Clearing this bit has no effect until the next disable event.			
1:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

9.2.5.8 HFXO_EXTCLKPKDETCTRL - Peak Detector Control Register

Offset	Bit Position																															
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x5				0x5			
Access																									RW				RW			
Name																									PKDETTHEXTCLK1				PKDETTHEXTCLK0			

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:4	PKDETTHEXTCLK1	0x5	RW	**PKDET Threshold1 EXTCLKPKDET mode**
	PKDET Threshold0 EXTCLKPKDET mode			
3:0	PKDETTHEXTCLK0	0x5	RW	**PKDET Threshold0 EXTCLKPKDET mode**
	PKDET Threshold0 EXTCLKPKDET mode			

9.2.5.9 HFXO_CMD - Command Register

Offset	Bit Position																															
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														W(nB)	W(nB)	W(nB)
Name																														STOPMEAS	STARTMEAS	COREBIASOPT

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	STOPMEAS	0x0	W(nB)	**Stop measuring crystal startup time**
	Stop measuring crystal startup time and clear associated registers			
1	STARTMEAS	0x0	W(nB)	**Start measuring crystal startup time**
	Start measuring crystal startup time			
0	COREBIASOPT	0x0	W(nB)	**Core Bias Optimizaton**
	Starts the core bias current optimization algorithm and runs it one time. Optimization should be executed if the temperature changes by more than 40degC. Do not run this command while the radio is in RX or TX modes. Do not issue this command more than once until COREBIASOPTRDY is asserted, or the previous command may be cancelled.			

9.2.5.10 HFXO_STATUS - Status Register

Offset	Bit Position																														
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x0						0x0		0x0	0x0		0x0	0x0							0x0	0x0				0x0		0x0	0x0	0x0
Access	R	R	R	R						R		R	R		R	R							R	R				R		R	R	R
Name	LOCK	SYNCBUSY	INJBSY	STUPMEASBSY						SYSRTCHWREQ		PRSHWREQ	ISWARM		HWREQ	ENS							SLEEPYXTALERR	SLEEPYXTAL				SYSRTCRDY		PRSRDY	COREBIASOPTRDY	RDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**Configuration Lock Status**
	Indicates the current status of configuration lock.			
	Value	Mode		Description
	0	UNLOCKED		Configuration lock is unlocked
	1	LOCKED		Configuration lock is locked
30	SYNCBUSY	0x0	R	**Sync Busy**
	Indicates synchronization is ongoing.			
29	INJBSY	0x0	R	**Clock Injection Busy**
	Clock Injection Busy			
28	STUPMEASBSY	0x0	R	**Startup Measurement Busy**
	Startup Measurement Busy			
27:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	SYSRTCHWREQ	0x0	R	**Oscillator Requested by SYSRTC Request**
	Oscillator is requested by the SYSRTC oscillator request interface.			
21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	PRSHWREQ	0x0	R	**Oscillator Requested by PRS Request**
	Oscillator is requested by the PRS oscillator request interface.			
19	ISWARM	0x0	R	**Oscillator Is Kept Warm**
	Oscillator is currently kept in warm state. Re-enable from warm state skips startup sequence.			
18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	HWREQ	0x0	R	**Oscillator Requested by Digital Clock**
	Oscillator is requested by digital clock request interface.			
16	ENS	0x0	R	**Enabled Status**
	Oscillator is enabled.			
15:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	SLEEPYXTALERR	0x0	R	**XTAL Startup Failed with Sleepy Settings**

Bit	Name	Reset	Access	Description
	XTAL Startup Failed with Sleepy Settings			
8	SLEEPYXTAL	0x0	R	**Sleepy XTAL settings applied**
	Sleepy XTAL settings applied			
7:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	SYSRTCRDY	0x0	R	**SYSRTC Ready Status**
	The SYSRTC oscillator startup is ready			
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	PRSRDY	0x0	R	**PRS Ready Status**
	The PRS oscillator startup is ready.			
1	COREBIASOPTRDY	0x0	R	**Core Bias Optimization Ready**
	Core bias current optimization algorithm is complete.			
0	RDY	0x0	R	**Ready Status**
	The digital clock branch (osc.clk_qual) is ready.			

9.2.5.11 HFXO_AVGSTARTUPTIME - Average Startup Time Register

Offset	Bit Position																															
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0								
Access																								R								
Name																								AVGSTUP								

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	AVGSTUP	0x0	R	**Average Startup Time**
	Average Startup Time averaged over 1, 8 or 16 measurements as configured in CFG.NUMSTUPMEAS			

9.2.5.12 HFXO_IF - Interrupt Flag Register

Offset	Bit Position																													
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0			0x0	0x0			0x0		0x0								0x0			0x0	0x0				0x0		0x0	0x0	0x0
Access	RW	RW	RW			RW	RW			RW		RW								RW			RW	RW				RW		RW	RW	RW
Name	COREBIASOPTERR	LFTIMEOUTERR	DNSERR			DNSSLEEPYXTALERR	INJERR			SYSRTCERR		PRSERR								INJSKIP			SLEEPYXTAL	STUPMEASDONE				SYSRTCRDY		PRSRDY	COREBIASOPTRDY	RDY

Bit	Name	Reset	Access	Description
31	COREBIASOPTERR	0x0	RW	**Core Bias Optimization Error Interrupt**
	Core bias current optimization algorithm fails to complete.			
30	LFTIMEOUTERR	0x0	RW	**Low Frequency Timeout Error Interrupt**
	Low frequency timeout triggers before the steady state timeout triggers.			
29	DNSERR	0x0	RW	**Did Not Start Error Interrupt**
	Crystal oscillator fails to startup.			
28:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26	DNSSLEEPYXTALERR	0x0	RW	**Did Not Start Sleepy Xtal Error IRQ**
	Startup Failed with Sleepy xtal settings.			
25	INJERR	0x0	RW	**Injection Error Interrupt**
	Timeout occurred during injection phase of startup indicating injection clock was not available			
24:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	SYSRTCERR	0x0	RW	**SYSRTC Request Error Interrupt**
	SYSRTC request is asserted while oscillator is forced to shutdown			
21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	PRSERR	0x0	RW	**PRS Requset Error Interrupt**
	PRS request is asserted while oscillator is forced to shutdown.			
19:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	INJSKIP	0x0	RW	**Injection Skipped**
	Injection was skipped for most recent startup because of invalid entry in SYVCOCAP LUT for current temperature			
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	SLEEPYXTAL	0x0	RW	**Sleepy crystal settings applied**
	Sleepy crystal settings applied since crystal failed successive attempts at startup with nominal settings			
8	STUPMEASDONE	0x0	RW	**Startup Time Measurement Complete**
	Startup Time Measurement and averaging complete, average startup time available in STATUS register			

Bit	Name	Reset	Access	Description
7:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	SYSRTCRDY	0x0	RW	**SYSRTC Ready Interrupt**
	The SYSRTC oscillator startup is ready			
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	PRSRDY	0x0	RW	**PRS Ready Interrupt**
	The PRS oscillator startup is ready.			
1	COREBIASOPTRDY	0x0	RW	**Core Bias Optimization Ready Interrupt**
	Core bias current optimization algorithm is complete.			
0	RDY	0x0	RW	**Digital Clock Ready Interrupt**
	The digital clock branch (osc.clk_qual) is ready.			

9.2.5.13 HFXO_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0			0x0	0x0			0x0		0x0								0x0			0x0	0x0				0x0		0x0	0x0	0x0
Access	RW	RW	RW			RW	RW			RW		RW								RW			RW	RW				RW		RW	RW	RW
Name	COREBIASOPTERR	LFTIMEOUTERR	DNSERR			DNSSLEEPYXTALERR	INJERR			SYSRTCERR		PRSERR								INJSKIP			SLEEPYXTAL	STUPMEASDONE				SYSRTCRDY		PRSRDY	COREBIASOPTRDY	RDY

Bit	Name	Reset	Access	Description
31	COREBIASOPTERR	0x0	RW	**Core Bias Optimization Error Interrupt**
	Core bias current optimization algorithm fails to complete.			
30	LFTIMEOUTERR	0x0	RW	**Low Frequency Timeout Error Interrupt**
	Low frequency timeout triggers before the steady state timeout triggers.			
29	DNSERR	0x0	RW	**Did Not Start Error Interrupt**
	Crystal oscillator fails to startup.			
28:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26	DNSSLEEPYXTALERR	0x0	RW	**Did Not Start Sleepy Xtal Error IRQ**
	Startup Failed with Sleepy Xtal settings.			
25	INJERR	0x0	RW	**Injection Error Interrupt**
	Timeout occurred during injection phase of startup indicating injection clock was not available			
24:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	SYSRTCERR	0x0	RW	**SYSRTC Request Error Interrupt**
	SYSRTC request is asserted while oscillator is forced to shutdown			
21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	PRSERR	0x0	RW	**PRS Requset Error Interrupt**
	PRS request is asserted while oscillator is forced to shutdown.			
19:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	INJSKIP	0x0	RW	**Injection Skipped**
	Injection was skipped for most recent startup because of invalid entry in SYVCOCAP LUT for current temperature			
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	SLEEPYXTAL	0x0	RW	**Sleepy crystal settings applied**
	Sleepy crystal settings applied since crystal failed successive attempts at startup with nominal settings			
8	STUPMEASDONE	0x0	RW	**Startup Time Measurement Complete**
	Startup Time Measurement and averaging complete, average startup time available in STATUS register			

Bit	Name	Reset	Access	Description
7:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	SYSRTCRDY	0x0	RW	**SYSRTC Ready Interrupt**
	The SYSRTC oscillator startup is ready			
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	PRSRDY	0x0	RW	**PRS Ready Interrupt**
	The PRS oscillator startup is ready.			
1	COREBIASOPTRDY	0x0	RW	**Core Bias Optimization Ready Interrupt**
	Core bias current optimization algorithm is complete.			
0	RDY	0x0	RW	**Digital Clock Ready Interrupt**
	The digital clock branch (osc.clk_qual) is ready.			

9.2.5.14 HFXO_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x080	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x580E															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x580E	W	**Configuration Lock Key**
	Write any other value than the unlock code to lock registers from editing. Write the unlock code to unlock.			

Value	Mode	Description
22542	UNLOCK	Write this value to unlock

9.3 SOCPLL - SoC Phase-Locked-Loop

9.3.1 Introduction

The SOCPLL is a fractional-N PLL that generates high frequency clocks for the system. It accepts an input frequency of 34 to 44 MHz and produces an output of 150 MHz to drive the main core SYSCLK. The SOCPLL may also be used in an open-loop mode to generate a 145 MHz clock for the system without using a reference oscillator.

9.3.2 Features

- Fractional-N PLL oscillator
- Fast startup and lock time
- Open-loop mode
- On-demand clock support
- Low output jitter
- Ready interrupt
- Lock loss interrupt

9.3.3 Functional Description

9.3.3.1 PLL Operation

The PLL function can be configured for integer or fractional-N mode. The CTRL.ENFRACN bit determines which mode will be used when the PLL is active. The reference clock for the PLL is selected using CTRL.REFCLKSEL. The local oscillator operates at 6 times the output frequency.

In fractional-N mode, the PLL provides a fractional multiple of the reference clock, determined by CTRL.DIVN and CTRL.DIVF. DIVN + 2 determines the integer portion of the multiplier, while DIVF / 1024 is the fractional portion. The output frequency is F_{REF} * (DIVN + 2 + (DIVF / 1024)) / 6.

If no fractional portion is required (i.e. DIVF = 0), the SOCPLL can operate in integer mode. CTRL.ENFRACN should be cleared to '0' when DIVF is 0. The output frequency equation reduces to F_{REF} * (DIVN + 2) / 6 in integer mode.

9.3.3.2 Start-up

The SOCPLL starts up very quickly. Refer to the device data sheet for start-up time specifications. After start-up, the RDY bit in the STATUS register will go high and the optional RDY interrupt will be triggered. The RDY signal is also used to indicate to downstream clock consumers that the clock is ready to use. The RDY flag can be delayed by 4, 8, or 16 us after PLL lock using the CTRL.PROGRDY field.

9.3.3.3 On-Demand Clocking

Software can request to enable the SOCPLL by setting the SOCPLL_CTRL.FORCEEN bit field. By default the SOCPLL is configured for on-demand clocking, which allows hardware to enable and disable the oscillator as necessary. This means that SOCPLL receives a request for clock from the CMU whenever the oscillator clock is needed. The on-demand mode can be disabeld by setting SOCPLL_CTRL.DISONDEMAND.

9.3.3.4 Open Loop Operation

The SOCPLL supports running in open-loop mode at a calibrated frequency of 145 MHz +/- 3%. Open-loop mode does not depend on other oscillator reference sources, and may provide some energy savings at the cost of accuracy. The SOCPLL operates in open-loop mode by setting CTRL.ENOPENLOOP to '1'.

9.3.4 SOCPLL Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	SOCPLL_IPVERSION	R	IP VERSION Register
0x004	SOCPLL_CTRL	RW SYNC	Control Register
0x018	SOCPLL_STATUS	RH SYNC	Status Register
0x01C	SOCPLL_IF	RWH INTFLAG	Interrupt Flag Register
0x020	SOCPLL_IEN	RW	Interrupt Enable Register
0x024	SOCPLL_LOCK	W	Configuration Lock Register
0x1000	SOCPLL_IPVERSION_SET	R	IP VERSION Register
0x1004	SOCPLL_CTRL_SET	RW SYNC	Control Register
0x1018	SOCPLL_STATUS_SET	RH SYNC	Status Register
0x101C	SOCPLL_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1020	SOCPLL_IEN_SET	RW	Interrupt Enable Register
0x1024	SOCPLL_LOCK_SET	W	Configuration Lock Register
0x2000	SOCPLL_IPVERSION_CLR	R	IP VERSION Register
0x2004	SOCPLL_CTRL_CLR	RW SYNC	Control Register
0x2018	SOCPLL_STATUS_CLR	RH SYNC	Status Register
0x201C	SOCPLL_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2020	SOCPLL_IEN_CLR	RW	Interrupt Enable Register
0x2024	SOCPLL_LOCK_CLR	W	Configuration Lock Register
0x3000	SOCPLL_IPVERSION_TGL	R	IP VERSION Register
0x3004	SOCPLL_CTRL_TGL	RW SYNC	Control Register
0x3018	SOCPLL_STATUS_TGL	RH SYNC	Status Register
0x301C	SOCPLL_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3020	SOCPLL_IEN_TGL	RW	Interrupt Enable Register
0x3024	SOCPLL_LOCK_TGL	W	Configuration Lock Register

9.3.5 SOCPLL Register Description

9.3.5.1 SOCPLL_IPVERSION - IP VERSION Register

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x2																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x2	R	**IP version register**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

9.3.5.2 SOCPLL_CTRL - Control Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x2				0x0		0x276													0x16							0x0		0x0	0x1	0x0	0x0
Access	RW				RW		RW													RW							RW		RW	RW	RW	RW
Name	PROGRDY				ENSOCCLK0		DIVF													DIVN							REFCLKSEL		ENOPENLOOP	ENFRACN	DISONDEMAND	FORCEEN

Bit	Name	Reset	Access	Description
31:30	PROGRDY	0x2	RW	**Programmable RDY delay**
	Programmable delay option after the pll lock to have SOCPLL finally ready for use.			

Value	Mode	Description
0	RDY_DELAY_0US	0us delay for rdy generation
1	RDY_DELAY_4US	4us delay for rdy generation
2	RDY_DELAY_8US	8us delay for rdy generation
3	RDY_DELAY_16US	16us delay for rdy generation

29:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	ENSOCCLK0	0x0	RW	**Enable flash phy clock**
	Enable clock output for QSPI / FPLL			
26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:16	DIVF	0x276	RW	**Fractional divider value**
	Fractional divider for frac-N operation. Set to 0 when operating in integer mode			
15:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12:8	DIVN	0x16	RW	**Integer divider value**
	Integer divider			
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	REFCLKSEL	0x0	RW	**Refrence clock source selection**
	Select/request reference clock			

Value	Mode	Description
0	REF_HFXO	Select HFXO as reference clock
1	REF_HFRCO	Select HFRCO as reference clock
2	REF_EXTCLK	Select external clock as reference clock

3	ENOPENLOOP	0x0	RW	**Open loop mode enable**
	Enable open loop mode			
2	ENFRACN	0x1	RW	**Frac-N Mode Enable**

Bit	Name	Reset	Access	Description
	Operate in integer mode when 0, frac-N mode when 1.			
1	DISONDEMAND	0x0	RW	**Disable on Demand**
	Disabling the on-demand request			
0	FORCEEN	0x0	RW	**Force Enable**
	Force enabling the gp_socpll block			

9.3.5.3 SOCPLL_STATUS - Status Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																							0x0							0x0	0x0
Access	R																							R							R	R
Name	LOCK																							ENS							PLLLOCK	RDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**Lock Status**
	If set, all SOCPLL lockable registers are locked.			

Value	Mode	Description
0	UNLOCKED	SOCPLL is unlocked
1	LOCKED	SOCPLL is locked

30:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	ENS	0x0	R	**SOCPLL Enable**
	SOCPLL is enabled			
7:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	PLLLOCK	0x0	R	**Goes high within TDC range**
	PLL_Lock status			
0	RDY	0x0	R	**SOCPLL Ready**
	SOCPLL is ready to use			

9.3.5.4 SOCPLL_IF - Interrupt Flag Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															LOSSOFLOCK	RDY

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	LOSSOFLOCK	0x0	RW	**Loss of PLL Lock Flag**
	Set when PLL lock is lost			
0	RDY	0x0	RW	**Ready Interrupt Flag**
	Set when SOCPLL is ready to use			

9.3.5.5 SOCPLL_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															LOSSOFLOCK	RDY

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	LOSSOFLOCK	0x0	RW	**LOSSOFLOCK Interrupt Enable**
	Enable/disable the LOSSOFLOCK interrupt			
0	RDY	0x0	RW	**RDY Interrupt Enable**
	Enable/disable the RDY interrupt			

9.3.5.6 SOCPLL_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x81A6															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x81A6	W	**Configuration Lock Key** Write any other value than the unlock code to lock registers from editing. Write the unlock code to unlock.

Value	Mode	Description
33190	UNLOCK	Write this value to unlock

9.4 HFRCO - High-Frequency RC Oscillator

9.4.1 Introduction

The HFRCO is a calibrated internal High Frequency RC oscillator.

9.4.2 Features

- Low start-up time
- Run-time band change or tuning

9.4.3 Functional Description

9.4.3.1 Start-up

The HFRCO starts up very quickly. Refer to the device data sheet for start-up time specifications. After the start-up time, the RDY status bit will go high and the RDY interrupt will be triggered. It can take another two clock cycles for the clock to propagate through the CMU before the clock is seen by peripherals.

9.4.3.2 On-Demand Clocking

Software can request to enable the HFRCO by setting the HFRCO_CTRL.FORCEEN bit field. The HFRCO can also optionally be configured via the HFRCO_CTRL.DISONDEMAND to shut down when no hardware request is present. This is known as on-demand clocking and allows the oscillator to be controlled without any software intervention. This means that HFRCO receives a request for clock from the CMU whenever the oscillator clock is needed. These requests can come at any time from any power domain (depending on the which peripheral is requesting the clock.)

9.4.3.3 Calibration

The HFRCO block supports a wide range of frequencies, and multiple HFRCO blocks may be included on any given device. The HFRCO_CAL register for each instance is used to control the oscillator characteristics and set the frequency.

The primary HFRCO, referred to as HFRCO0 or HFRCODPLL in this document, has two production-calibrated conditions for specific open-loop frequencies stored in the Device Info space: HFRCOCALDEFAULT, and HFRCOCALSPEED. After a reset, the HFRCO0_CAL register will be set to the HFRCOCALDEFAULT value.

Additionally, HFRCO0 / HFRCODPLL can use the DPLL block to produce a more accurate output frequency given a good reference source. The oscillator supports 11 bands (frequency ranges). Each band has a production-calibrated condition stored in the Device Info space: HFRCODPLLBAND0 through HFRCODPLLBAND10. The range for each band is shown in the device data sheet. Software should select the appropriate band containing the target frequency and program the band calibration setting (HFRCODPLLBANDx) into the CAL register before enabling DPLL to ensure the target frequency can be reached.

The second HFRCO, referred to as HFRCO1 or HFRCOEM23 in this document, has a single production-calibrated condition for a specific open-loop frequency stored in the Device Info space: HFRCOEM23DEFAULT. After a reset, the HFRCOEM23_CAL register will be set to the HFRCOEM23DEFAULT value.

The production-calibrated settings may be used as-is to program the oscillator to the target frequency or range. The TUNING and FINETUNING bit fields in the CAL register can also be used to trim HFRCO manually.

Note:

When trimming the HFRCO frequency using TUNING and FINETUNING, the appropriate HFRCODPLLBANDx value should be written to the CAL register to establish the correct frequency range.

Software may write the CAL register at any time. If there is already a frequency updating occuring, the current change would apply when the previous update is done. FREQBSY in STATUS register indicates if the updating is finished.

To support lower frequencies than what is listed as the minimum HFRCODPLLBAND0 limit, an optional output divider is provided. The oscillator frequency remains in the specified band, but the clock exported to the rest of the device is divided by the setting programmed into CAL.CLKDIV. CLKDIV settings of 1, 2, 4, and 8 are supported.

9.4.3.4 Interrupts

HFRCO has one interrupt: IF.RDY. RDY is triggered when the timeout has finished and the qualified HFRCO clock is ready. The clock is gated until it is ready.

9.4.3.5 Status Flags

9.4.3.5.1 FREQBSY

The FREQBSY bit indicates the HFRCO is busy updating its frequency after a write to the CAL register. The FREQBSY bit should be polled whenever frequency is changed. After software writes to the CAL register, FREQBSY asserts immediately. Software should wait for FREQBSY to be zero before attempting to write to the CAL register again.

For a band change, FREQBSY will not de-assert until after the timeout upon being re-enabled.

For normal start-up, FREQBSY will not assert.

During DPLL operation, FREQBSY will not assert, as the frequency adjustments are not caused by writes to the CAL register.

When disabling DPLL, the last tuning value is written back to the CAL register, and FREQBSY will assert.

9.4.3.5.2 ENS

ENS indicates the HFRCO is enabled. This flag is used to check if the HFRCO is enabled by any requester.

Note: When a band change occurs, the HFRCO is disabled and re-enabled. This will cause the ENS bit to briefly de-assert.

9.4.3.5.3 RDY

RDY indicates HFRCO is enabled and start-up timeout has exceeded. Used to check if the HFRCO clock is ready after enable.

Changing bands will de-assert RDY as the oscillator must reset and start up again.

9.4.3.5.4 SYNCBUSY

SYNCBUSY indicates ongoing synchronization of CAL register fields after a write.

9.4.3.6 Forced Oscillator Control

The HFRCO can be forced on and off using the FORCEEN and DISONDEMAND bits in the CTRL register.

Setting FORCEEN will force the oscillator core to run, but peripherals will still need to request the clock to un-gate the clock signal.

9.4.3.7 Oscillator Modes

The HFRCO has three modes of operation, an **on-demand** mode (which is the normal software use case), a **force on** and a **force off** mode.

In **on-demand** mode the oscillator will start whenever a peripheral requests it. Which in most cases is whenever the peripheral is enabled.

In **force on** mode the analog core will run independently of whether it is requested or not. This can be useful for measuring analog current without any digital load on the clocks.

In **force off** mode, the analog core will be shut off independently of whether it is requested or not. This can be useful for changing analog test settings without risking glitches on the clock.

The DISONDEMAND bit can also be used to give software full control over the clock for exceptional cases where software control is desired.

Table 9.1. Oscillator modes

Bit Field	FORCEEN	DISONDEMAND
On-Demand (normal operation)	0	0
Forced On	1	X
Forced Off	0	1

9.4.4 HFRCO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	HFRCO_IPVERSION	R	IP Version ID
0x004	HFRCO_CTRL	RW	Ctrl Register
0x008	HFRCO_CAL	RWH SYNC	Calibration Register
0x00C	HFRCO_STATUS	RH	Status Register
0x010	HFRCO_IF	RWH INTFLAG	Interrupt Flag Register
0x014	HFRCO_IEN	RW	Interrupt Enable Register
0x01C	HFRCO_LOCK	W	Lock Register
0x1000	HFRCO_IPVERSION_SET	R	IP Version ID
0x1004	HFRCO_CTRL_SET	RW	Ctrl Register
0x1008	HFRCO_CAL_SET	RWH SYNC	Calibration Register
0x100C	HFRCO_STATUS_SET	RH	Status Register
0x1010	HFRCO_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1014	HFRCO_IEN_SET	RW	Interrupt Enable Register
0x101C	HFRCO_LOCK_SET	W	Lock Register
0x2000	HFRCO_IPVERSION_CLR	R	IP Version ID
0x2004	HFRCO_CTRL_CLR	RW	Ctrl Register
0x2008	HFRCO_CAL_CLR	RWH SYNC	Calibration Register
0x200C	HFRCO_STATUS_CLR	RH	Status Register
0x2010	HFRCO_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2014	HFRCO_IEN_CLR	RW	Interrupt Enable Register
0x201C	HFRCO_LOCK_CLR	W	Lock Register
0x3000	HFRCO_IPVERSION_TGL	R	IP Version ID
0x3004	HFRCO_CTRL_TGL	RW	Ctrl Register
0x3008	HFRCO_CAL_TGL	RWH SYNC	Calibration Register
0x300C	HFRCO_STATUS_TGL	RH	Status Register
0x3010	HFRCO_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3014	HFRCO_IEN_TGL	RW	Interrupt Enable Register
0x301C	HFRCO_LOCK_TGL	W	Lock Register

9.4.5 HFRCO Register Description

9.4.5.1 HFRCO_IPVERSION - IP Version ID

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x3																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x3	R	**IP Version**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

9.4.5.2 HFRCO_CTRL - Ctrl Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															DISONDEMAND	FORCEEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISONDEMAND	0x0	RW	**Disable On-demand**
	Setting this bit disable HFRCO on-demand feature			
0	FORCEEN	0x0	RW	**Force Enable**
	Setting this bit force HFRCO enabled			

9.4.5.3 HFRCO_CAL - Calibration Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xA				0x3			0x0		0x0		0x6						0x0			0x1F						0x1B					
Access	RW				RW			RW		RW		RW						RW			RW						RW					
Name	TEMPCO				BIAS			SIGMADELTA		CLKDIV		BANDFINE						BANDCOARSE			FINETUNING						TUNING					

Bit	Name	Reset	Access	Description
31:28	TEMPCO	0xA	RW	**Tempco Trim**
	Writing this field adjusts the temperature coefficient trim on comparator current.			
27:25	BIAS	0x3	RW	**Bias Current Setting**
	Writing this field adjusts the HFRCO comparator bias current.			
24:23	SIGMADELTA	0x0	RW	**Sigma Delta Setting**
22:21	CLKDIV	0x0	RW	**Locally Divide HFRCO Clock Output**
	Writing this field configures the HFRCO clock output divider.			

Value	Mode	Description
0	DIV1	Divide by 1.
1	DIV2	Divide by 2.
2	DIV4	Divide by 4.
3	DIV8	Divide by 8

20:15	BANDFINE	0x6	RW	**Band Fine Value**
14:12	BANDCOARSE	0x0	RW	**Band Coarse Value**
	Writing this field adjusts the HFRCO frequency range.			
11:6	FINETUNING	0x1F	RW	**Fine Tuning Value**
	Writing this field adjusts the HFRCO fine tuning value. Higher value means lower frequency.			
5:0	TUNING	0x1B	RW	**Coarse Tuning Value**
	Writing this field adjusts the HFRCO coarse tuning value. Higher value means lower frequency.			

9.4.5.4 HFRCO_STATUS - Status Register

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0															0x0														0x0	0x0	0x0
Access	R															R														R	R	R
Name	LOCK															ENS														SYNCBUSY	FREQBSY	RDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**Lock Status**
	If set, all HFRCO lockable registers are locked.			

Value	Mode	Description
0	UNLOCKED	HFRCO is unlocked
1	LOCKED	HFRCO is locked

30:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	ENS	0x0	R	**Enable Status**
	HFRCO is enabled.			
15:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	SYNCBUSY	0x0	R	**Synchronization Busy**
	This bit is set when there is an ongoing synchronization of CAL register bitfields.			
1	FREQBSY	0x0	R	**Frequency Updating Busy**
	HFRCO is busy updating frequency.			
0	RDY	0x0	R	**Ready**
	HFRCO is enabled and start-up time has exceeded.			

9.4.5.5 HFRCO_IF - Interrupt Flag Register

Offset	Bit Position																															
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																RDY

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	RDY	0x0	RW	**Ready Interrupt Flag**
	Set when HFRCO is ready (start-up time exceeded).			

9.4.5.6 HFRCO_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																RDY

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	RDY	0x0	RW	**RDY Interrupt Enable**
	Enable/disable the RDY interrupt			

9.4.5.7 HFRCO_LOCK - Lock Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x8195															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x8195	W	**Lock Key**
	Write any other value than the unlock code to lock registers from editing. Write the unlock code to unlock.			

Value	Mode	Description
33173	UNLOCK	Unlock code

9.5 DPLL - Digital Phased Locked Loop

9.5.1 Introduction

The Digital Phase-Locked Loop (DPLL) uses a reference clock to generate a desired clock frequency at a specified ratio to the reference clock.

9.5.2 Features

- Frequency Lock Mode
- Phase-Lock Mode
- Output frequency = F_{REF}*(N+1)/(M+1), where N and M are 12-bit values
- Very fast lock time
- Very fast transient tracking
- Low output jitter
- Lock detection with an interrupt
- Lock fail detection with interrupts

9.5.3 Functional Description

9.5.3.1 Enabling and Disabling

The DPLL can be enabled and disabled by software via the DPLL_EN register. Before enabling DPLL, software should:

1. Select reference clock by setting the CLKSEL field in CMU_DPLLREFCLKCTRL.
2. The CMU should not be running from the HFRCO. If necessary, the CMU should switch to the FSRCO until after the DPLL has locked to avoid over-clocking due to overshoot. If necessary, select FSRCO or HFXO in the CMU_SYSCLKCTRL register CLKSEL field.
3. Configure the DPLL.
4. Make certain that the ENS bit in DPLL_STATUS is low.

The DPLL is disabled automatically when entering energy modes lower than EM1. Note that disabling the DPLL will not automatically turn off the reference clock. The CLKSEL field in CMU_DPLLREFCLKCTRL must be set to DISABLED or the referenced clock may continue to run even when hardware has disabled the DPLL.

9.5.3.2 Lock Modes

The DPLL provides two lock modes, referred to as frequency-lock loop mode (FREQLL) and phase-lock loop mode (PHASELL). FREQLL mode keeps the DCO frequency-locked to the reference clock, which means the DCO frequency will be accurate. However, the phase error can accumulate over time and cause a non-zero average frequency error. FREQLL mode also provides better jitter and transient performance. PHASELL mode keeps the DCO phase-locked to the reference clock, which means the phase error does not accumulate over time, which makes the average frequency error zero. FREQLL mode is usually sufficient unless specific phase requirement exists.

9.5.3.3 Configurations

The formula for the DPLL output frequqncy is FREF*(N+1)/(M+1). The user should calculate N and M in DPLL_CFG1 to achieve the target frequency. Note that with a larger value of N, the DCO lock time would increase and DCO jitter would decrease. Both effects are approximately linear. This relationship can be used to select N for a given application to strike a compromise between lock time and output jitter. For example if an ratio of 3 is desired, the DPLL could be configured as {N=599, M=199} for fast lock time but high jitter, or as {N=2999, M=999} for lower jitter but longer lock time.

Note: All configuration settings should be done before enabling the DPLL. They should not be changed when DPLL is running. The final tuning values can be read back from TUNING and FINETUNING in HFRCO_CAL, after DPLL is disabled and DPLLENS in DPLL_STATUS is low.

9.5.3.4 Lock Detection

The DPLL has 3 different types of output events: ready, lock fail due to period underflow, and lock fail due to period overflow. Each of the events has its own interrupt flag. DPLLRDY is set when DPLL successfully locks to the reference clock based on the software configuration. DPLLLOCKFAILLOW is set when the DPLL fails to lock because the period lower boundary is hit. DPLLLOCKFAILHIGH is set when the DPLL fails to lock because the period upper boundary is hit. If the interrupt flags are set and the corresponding interrupt enable bits in DPLL_IEN are set, the DPLL will request an interrupt. Based on different interrupt events, software should take different actions:

- If the DPLLRDY interrupt is received first, it means target clock is ready and it is safe to switch to use DCO's output.
- If the DPLLLOCKFAILLOW interrupt is received first, it indicates the RANGE in HFRCO_CAL is too small. Software should disable the DPLL and write a larger value to RANGE, then enable the DPLL again to lock.
- If the DPLLLOCKFAILHIGH interrupt is received first, it indicates the RANGE in HFRCO_CAL is too large. Software should disable DPLL and write a smaller value to RANGE, then enable DPLL again to lock.
- If the DPLLRDY interrupt is received first and then DPLLLOCKFAILLOW or DPLLLOCKFAILHIGH is received later, it means reference clock drifted over 1% and the DPLL has lost its locked status.
  - If AUTORECOVER in DPLL_CFG is not set, software should disable the DPLL and enable DPLL again to lock.
  - If AUTORECOVER in DPLL_CFG is set, hardware will re-lock automatically. When the target frequency is near the boundary of a range, the drift may cause underflow or overflow. In this case the fail interrupt will still be received. Software should disable the DPLL and modify RANGE in HFRCO_CAL in corresponding direction, depending on whether the DPLLLOCKFAILLOW or DPLLLOCKFAILHIGH bit is set. Then enable DPLL again to lock.

9.5.4 DPLL Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	DPLL_IPVERSION	R	IP Version
0x004	DPLL_EN	RW ENABLE	Enable
0x008	DPLL_CFG	RW CONFIG	Config
0x00C	DPLL_CFG1	RW CONFIG	Config1
0x010	DPLL_IF	RWH INTFLAG	Interrupt Flag
0x014	DPLL_IEN	RW	Interrupt Enable
0x018	DPLL_STATUS	RH	Status
0x024	DPLL_LOCK	W	Lock
0x1000	DPLL_IPVERSION_SET	R	IP Version
0x1004	DPLL_EN_SET	RW ENABLE	Enable
0x1008	DPLL_CFG_SET	RW CONFIG	Config
0x100C	DPLL_CFG1_SET	RW CONFIG	Config1
0x1010	DPLL_IF_SET	RWH INTFLAG	Interrupt Flag
0x1014	DPLL_IEN_SET	RW	Interrupt Enable
0x1018	DPLL_STATUS_SET	RH	Status
0x1024	DPLL_LOCK_SET	W	Lock
0x2000	DPLL_IPVERSION_CLR	R	IP Version
0x2004	DPLL_EN_CLR	RW ENABLE	Enable
0x2008	DPLL_CFG_CLR	RW CONFIG	Config
0x200C	DPLL_CFG1_CLR	RW CONFIG	Config1
0x2010	DPLL_IF_CLR	RWH INTFLAG	Interrupt Flag
0x2014	DPLL_IEN_CLR	RW	Interrupt Enable
0x2018	DPLL_STATUS_CLR	RH	Status
0x2024	DPLL_LOCK_CLR	W	Lock
0x3000	DPLL_IPVERSION_TGL	R	IP Version
0x3004	DPLL_EN_TGL	RW ENABLE	Enable
0x3008	DPLL_CFG_TGL	RW CONFIG	Config
0x300C	DPLL_CFG1_TGL	RW CONFIG	Config1
0x3010	DPLL_IF_TGL	RWH INTFLAG	Interrupt Flag
0x3014	DPLL_IEN_TGL	RW	Interrupt Enable
0x3018	DPLL_STATUS_TGL	RH	Status
0x3024	DPLL_LOCK_TGL	W	Lock

9.5.5 DPLL Register Description

9.5.5.1 DPLL_IPVERSION - IP Version

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP Version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

9.5.5.2 DPLL_EN - Enable

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement Busy Status**
	When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFO			
0	EN	0x0	RW	**Module Enable**
	The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.			

9.5.5.3 DPLL_CFG - Config

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																										0x0				0x0	0x0	0x0
Access																										RW				RW	RW	RW
Name																										DITHEN				AUTORECOVER	EDGESEL	MODE

Bit	Name	Reset	Access	Description
31:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	DITHEN	0x0	RW	**Dither Enable Control**
	Set to enable dither function			
5:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	AUTORECOVER	0x0	RW	**Automatic Recovery Control**
	Set to enable automatic recovery function			
1	EDGESEL	0x0	RW	**Reference Edge Select**
	This bit controls which edge of reference is detected			
0	MODE	0x0	RW	**Operating Mode Control**
	This bit controls which mode DPLL is operating when enabled			

Value	Mode	Description
0	FLL	Frequency Lock Mode
1	PLL	Phase Lock Mode

9.5.5.4 DPLL_CFG1 - Config1

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																0x0											
Access					RW																RW											
Name					N																M										

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:16	N	0x0	RW	**Factor N**
	The locked DCO frequency is given by: Fdco = Fref * (N + 1)/(M+1). N is required to be larger than 300.			
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:0	M	0x0	RW	**Factor M**
	The locked DCO frequency is given by: Fdco = Fref * (N + 1)/(M+1). M can be any value.			

9.5.5.5 DPLL_IF - Interrupt Flag

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														LOCKFAILHIGH	LOCKFAILLOW	LOCK

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	LOCKFAILHIGH	0x0	RW	**Lock Failure High Interrupt Flag**
	Set when DPLL fail to lock because of period overflow.			
1	LOCKFAILLOW	0x0	RW	**Lock Failure Low Interrupt Flag**
	Set when DPLL fail to lock because of period underflow.			
0	LOCK	0x0	RW	**Lock Interrupt Flag**
	Set when DPLL achieve the lock.			

9.5.5.6 DPLL_IEN - Interrupt Enable

Offset	Bit Position																														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														LOCKFAILHIGH	LOCKFAILLOW	LOCK

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	LOCKFAILHIGH	0x0	RW	**LOCKFAILHIGH Interrupt Enable**
	LOCKFAILHIGH Interrupt Enable			
1	LOCKFAILLOW	0x0	RW	**LOCKFAILLOW Interrupe Enable**
	LOCKFAILLOW Interrupe Enable			
0	LOCK	0x0	RW	**LOCK interrupt Enable**
	LOCK interrupt Enable			

9.5.5.7 DPLL_STATUS - Status

Offset	Bit Position																														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																														0x0	0x0
Access	R																														R	R
Name	LOCK																														ENS	RDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**Lock Status**
	Indicates the current status of configuration lock			

Value	Mode	Description
0	UNLOCKED	DPLL is unlocked
1	LOCKED	DPLL is locked

30:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	ENS	0x0	R	**Enable Status**
	DPLL is enabled.			
0	RDY	0x0	R	**Ready Status**
	DPLL is enabled and locked.			

9.5.5.8 DPLL_LOCK - Lock

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x7102															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x7102	W	**Lock Key**

Write any other value than the unlock code to lock registers from editing. Write the unlock code to unlock.

Value	Mode	Description
28930	UNLOCK	Unlock code

9.6 LFXO - Low-Frequency Crystal Oscillator

9.6.1 Introduction

The Low Frequency Crystal Oscillator (LFXO) uses an external 32.768 kHz crystal to provide an accurate low-frequency clock. The module is available in energy modes EM0, EM1, and EM4. The main interaction is with the CMU through the clock requesting mechanism.

9.6.2 Features

High-level features.

- Crystal calibration
- Functional in energy modes EM0, EM1, and EM4
- Failure detection and EM4 wakeup
- External sine input mode
- External CMOS input mode
- Edge interrupts
- On-demand oscillator enabling

9.6.3 Functional Description

9.6.3.1 Modes

The LFXO can be used in multiple different modes. The mode can be programmed by setting the MODE bit field in the LFXO_CFG register. If MODE is set to XTAL, the LFXO is programmed to operate in crystal mode and a 32.768 kHz crystal oscillator should be connected to the pins assigned to the LF crystal signals, LFXO.LFXTAL_I and LFXO.LFXTAL_O. If MODE is set to BUFEXTCLK, the LFXO is programmed to operate in external sine mode and the sine wave should be supplied to the pin assigned to LFXO.LF_EXTCLK. If MODE is set to DIGEXTCLK, LFXO is programmed to operate in external CMOS mode and the external 32.768 kHz clock should be provided on the LFXO.LF_EXTCLK input. For BUFEXTCLK and DIGEXTCLK modes, the LFXO.LFXTAL_O signal is not required, and does not need to be enabled on the GPIO.

9.6.3.2 Enabling

There are two ways to turn on the LFXO clock. One is to turn it on in FORCEON mode by setting FORCEEN bit to 1 in LFXO_CTRL register. Another is to keep it ready to be turned on in ONDEMAND mode by setting FORCEEN bit to 0 and DISONDEMAND bit to 0 in LFXO_CTRL register. This means that the oscillator will be off unless its clock requested. When a peripheral requests the clock, hardware will automatically enable the LFXO without any software intervention. The oscillator will remain on as long as the peripheral requests it. The DISONDEMAND setting does not have any impact when FORCEEN set to 1. LFXO is in FORCEOFF mode when FORCEEN set to 0 and DISONDEMAND set to 1. In FORCEOFF mode all requests are blocked and LFXO will not generate the clock.

9.6.3.3 Clock Qualification

Once the LFXO is enabled, the clock should not be used until it has had time to stabilize. Therefore, a number of cycles are required to qualify the clock. Before the clock is qualified, no clock requesters will receive the LFXO clock. The number of cycles used to qualify the clock can be programmed by setting the TIMEOUT bit field in the LFXO_CFG register. The TIMEOUT default value is set to 32,728 cycles, which is much more than necessary for stabilization. The stabilization time required will depend on the particular crystal, oscillator settings, and frequency accuracy requirements. A value of 4096 clocks is generally recommended for most applications. A low timeout of 2 cycles may be used in DIGEXTCLK mode in order to filter out the first glitch from the pin. The 2 clock cycle timeout should not be used with crystals. There are two status bits and one interrupt associated with enabling the oscillator and qualifying its clock. Once the oscillator gets enabled the ENS bit in LFXO_STATUS register will be set high. Note that due to the nature of on demand clocking, the oscillator can be enabled anytime, so if software reads ENS low it is not safe to assume that ENS stays low during the next instruction. It is only safe to assume that oscillator was OFF at the time ENS was read. Similarly, if software reads ENS high it is not safe to assume that ENS stays high during the next instruction. Once the clock is qualified, the RDY status is set high in the LFXO_STATUS register. The same uncertainties also apply to the RDY bit. However, software can wait for the RDY bit to go high to detect that the LFXO clock is qualified. Alternatively it can enable the ready interrupt with RDYIEN in LFXO_IEN register and receive RDYIF interrupt available in LFXO_IF register. RDYIF also acts as a wakeup source if RDYIEN set high. If put into FORCEON mode, the LFXO will start the qualification and once qualified it will gate off the clock but immediately start with no qualification upon receiving a request. If in ONDEMAND mode, the LFXO re-starts the qualification every time it is switched from off to on due to clock requests. Note that only enabling the RDY interrupt does not act as a clock request.

9.6.3.4 Edge Detection Interrupts

There is a possibility for software to detect rising or falling edges of the LFXO clock. The edge detection is enabled if any of POSEDGEIEN and NEGEDGEIEN is set to 1. The corresponding flags are available in POSEDGEIF and NEGEDGEIF. If none of the interrupts are enabled, the edge detection is disabled and POSEDGEIF and NEGEDGEIF hold their last value until cleared or set by software. Disabling the edge detection is only allowed on NEGEDGEIF. Both flags act as wakeup sources if the corresponding IEN is set high.

9.6.3.5 Clock Failure

In case the oscillator or crystal stops or does not output a clock when expected, a failure interrupt can be raised. The failure occurs if fewer than 3 LFXO clock positive edges happen during a 1 ms interval. Failure detection is enabled by setting FAILDETEN to 1 in LFXO_CTRL register. This bit acts as a clock requester. Once enabled, failure detection status can be checked by reading FAILIF in LFXO_IF register. If FAILIEN is set high, failure will generate an interrupt. Failure detection is also available as an EM4 wakeup source. To wakeup from EM4 on LFXO failure detection, set FAILDETEM4WUEN high in LFXO_CTRL.

9.6.3.6 Automatic Gain Control

The LFXO circuit includes optional AGC features. AGC is enabled by setting the DRVCURR field in LFXO_CFG to AGC. The AGC circuit has three amplitude target levels, controlled by LFXO_CFG.REFLEVEL. When active, the AGC circuit will make periodic adjustments to the driver strength to maintain the desired amplitude. There are also three settings for how often the AGC evaulates and adjusts the driver, controlled by LFXO_CFG.AGCRATE.

Setting the LFXO_CFG.DRVCURR field to a value other than AGC will disable the AGC function and force the oscillator to use the DRVCURR setting. If AGC is not used, LFXO_AGCCTRL.DISAGCCLK may be set to disable the clock for the AGC logic.

9.6.3.7 Force Off

It is not allowed to write to LFXO_CFG unless LFXO is in FORCEOFF mode. If this guideline is violated, the write access is blocked and a bus fault is generated. Writing to CFG registers has no effect in DIGEXTCLK mode. When putting the oscillator to FORCEOFF mode, software should wait for the ENS status bit to go low to indicate that the oscillator is shut off. Once the oscillator is off, it is safe to write to the LFXO_CFG register.

9.6.3.8 Register Synchronization

While the CFG registers are static LFXO configuration, LFXO_CAL register has GAIN and CAPTUNE bit fields which can be written to while the oscillator is running. This is used to calibrate the LFXO clock. These registers are allowed to be written only if CALBSY in LFXO_SYNCBUSY register is low. If this guideline is violated, the write access is blocked and a bus fault is generated. CALBSY is guaranteed to be low in FORCEOFF mode. When exiting FORCEOFF mode, CALBSY will go high and stay high until the initial internal synchronization is done. CALBSY is also guaranteed to be low in DIGEXTCLK mode since writing to CAL register has no effect in DIGEXTCLK mode. CAPTUNE is allowed to be incremented or decremented by one LSB when not in FORCEOFF mode. Note that CAPTUNE tunes the internal capacitors connected to LFXO.LFXTAL_I and LFXO.LFXTAL_O. By programming GAIN bit field it is possible to optimize start-up time and power consumption for a given crystal.

9.6.3.9 Register Lock

See the LFXO_LOCK register on how to lock certain registers. Registers LFXO_CTRL, LFXO_CFG, and LFXO_CAL are lockable. The LOCK bit in LFXO_STATUS register is available to check whether the registers are locked. If locked, all updates to these registers are blocked and bus faults are issued.

9.6.3.10 Reset Behavior

Upon reset, the LFXO is configured for the safe crystal start-up. The TIMEOUT is set to 32k cycles, The MODE is set to XTAL and the reset state is FORCEOFF. In order to minimize the start-up time and power consumption for a given crystal, it is possible to adjust the start-up gain in the oscillator by programming GAIN in LFXO_CAL. All controls are retained in EM4, except the LFXO_IEN register which is reset after EM4 wakeup.

9.6.4 LFXO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LFXO_IPVERSION	R	LFXO IP Version
0x004	LFXO_CTRL	RW	LFXO Control Register
0x008	LFXO_CFG	RW	LFXO Configuration Register
0x010	LFXO_STATUS	RH	LFXO Status Register
0x014	LFXO_CAL	RW LFSYNC	LFXO Calibration Register
0x018	LFXO_IF	RWH INTFLAG	Interrupt Flag Register
0x01C	LFXO_IEN	RW	Interrupt Enable Register
0x020	LFXO_SYNCBUSY	RH	LFXO Sync Busy Register
0x024	LFXO_LOCK	W	Configuration Lock Register
0x02C	LFXO_AGCCTRL	RW	AGC Control Register
0x1000	LFXO_IPVERSION_SET	R	LFXO IP Version
0x1004	LFXO_CTRL_SET	RW	LFXO Control Register
0x1008	LFXO_CFG_SET	RW	LFXO Configuration Register
0x1010	LFXO_STATUS_SET	RH	LFXO Status Register
0x1014	LFXO_CAL_SET	RW LFSYNC	LFXO Calibration Register
0x1018	LFXO_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x101C	LFXO_IEN_SET	RW	Interrupt Enable Register
0x1020	LFXO_SYNCBUSY_SET	RH	LFXO Sync Busy Register
0x1024	LFXO_LOCK_SET	W	Configuration Lock Register
0x102C	LFXO_AGCCTRL_SET	RW	AGC Control Register
0x2000	LFXO_IPVERSION_CLR	R	LFXO IP Version
0x2004	LFXO_CTRL_CLR	RW	LFXO Control Register
0x2008	LFXO_CFG_CLR	RW	LFXO Configuration Register
0x2010	LFXO_STATUS_CLR	RH	LFXO Status Register
0x2014	LFXO_CAL_CLR	RW LFSYNC	LFXO Calibration Register
0x2018	LFXO_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x201C	LFXO_IEN_CLR	RW	Interrupt Enable Register
0x2020	LFXO_SYNCBUSY_CLR	RH	LFXO Sync Busy Register
0x2024	LFXO_LOCK_CLR	W	Configuration Lock Register
0x202C	LFXO_AGCCTRL_CLR	RW	AGC Control Register
0x3000	LFXO_IPVERSION_TGL	R	LFXO IP Version
0x3004	LFXO_CTRL_TGL	RW	LFXO Control Register
0x3008	LFXO_CFG_TGL	RW	LFXO Configuration Register
0x3010	LFXO_STATUS_TGL	RH	LFXO Status Register
0x3014	LFXO_CAL_TGL	RW LFSYNC	LFXO Calibration Register
0x3018	LFXO_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x301C	LFXO_IEN_TGL	RW	Interrupt Enable Register
0x3020	LFXO_SYNCBUSY_TGL	RH	LFXO Sync Busy Register

Offset	Name	Type	Description
0x3024	LFXO_LOCK_TGL	W	Configuration Lock Register
0x302C	LFXO_AGCCTRL_TGL	RW	AGC Control Register

9.6.5 LFXO Register Description

9.6.5.1 LFXO_IPVERSION - LFXO IP Version

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x2																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x2	R	**IP Version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

9.6.5.2 LFXO_CTRL - LFXO Control Register

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0			0x1	0x0
Access																											RW	RW			RW	RW
Name																											FAILDETEM4WUEN	FAILDETEN			DISONDEMAND	FORCEEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	FAILDETEM4WUEN	0x0	RW	**LFXO Failure Detection EM4WU Enable**
	Set this bit to enable EM4 exit on the oscillator failure detection.			
4	FAILDETEN	0x0	RW	**LFXO Failure Detection Enable**
	Set this bit to enable the oscillator failure detection feature. Note that setting this bit will enable the oscillator core.			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISONDEMAND	0x1	RW	**LFXO Disable On-demand requests**
	Set this bit to disable On-demand requests.			
0	FORCEEN	0x0	RW	**LFXO Force Enable**
	Set this bit to enable the oscillator core. The oscillator core is enabled regardless of On-demand requests.			

9.6.5.3 LFXO_CFG - LFXO Configuration Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x1		0x0					0x7					0x0				0x0	
Access															RW		RW					RW					RW				RW	
Name															REFLEVEL		DRVCURR					TIMEOUT					MODE				AGCRATE	

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	REFLEVEL	0x1	RW	**LFXO Reference level for AGC**
	Set the target amplitude for the AGC			

Value	Mode	Description
0	MIN	low agc amplitude
1	RESET	default
3	MAX	high agc amplitude

Bit	Name	Reset	Access	Description
15:12	DRVCURR	0x0	RW	**LFXO Driver Current Setting**
	Manual driver current setting. AGC if set to 0			

Value	Mode	Description
0	AGC	
1	FORCE_1	
2	FORCE_2	
3	FORCE_3	
4	FORCE_4	
5	FORCE_5	
6	FORCE_6	
7	FORCE_7	
8	FORCE_8	
9	FORCE_9	
10	FORCE_10	
11	FORCE_11	
12	FORCE_12	
13	FORCE_13	
14	FORCE_14	
15	FORCE_15	

Bit	Name	Reset	Access	Description
11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:8	TIMEOUT	0x7	RW	**LFXO Start-up Delay**
	Configures the start-up delay for LFXO.			

Bit	Name	Reset	Access	Description

Value	Mode	Description
0	CYCLES2	Timeout period of 2 cycles
1	CYCLES256	Timeout period of 256 cycles
2	CYCLES1K	Timeout period of 1024 cycles
3	CYCLES2K	Timeout period of 2048 cycles
4	CYCLES4K	Timeout period of 4096 cycles
5	CYCLES8K	Timeout period of 8192 cycles
6	CYCLES16K	Timeout period of 16384 cycles
7	CYCLES32K	Timeout period of 32768 cycles

Bit	Name	Reset	Access	Description
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	MODE	0x0	RW	**LFXO Mode**

Selects the LFXO mode.

Value	Mode	Description
0	XTAL	LFXO in Low Power mode. A 32768 Hz crystal should be connected to the LF crystal pads.
1	XTAL_HP	LFXO in High Power mode. A 32768 Hz crystal should be connected to the LF crystal pads.
2	BUFEXTCLK	An external sine source should be connected to LFXTAL_I pin. The input signal is internally AC coupled. Voltage should be kept above ground and below DVDD. LFXTAL_O pin is free to use as a GPIO.
3	DIGEXTCLK	An external 32 kHz CMOS clock should be provided on LFXTAL_I. LFXTAL_O is free to be used as a general purpose GPIO.

Bit	Name	Reset	Access	Description
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	AGCRATE	0x0	RW	**LFXO AGC Refresh Rate**

Set how often the AGC will evaluate crystal amplitude and adjust driving power

Value	Mode	Description
0	SLOW	128k cycles
1	MEDIUM	32k cycles
2	FAST	8k cycles
3	TEST	RESERVED

9.6.5.4 LFXO_STATUS - LFXO Status Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0															0x0																0x0
Access	R															R																R
Name	LOCK															ENS																RDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**LFXO Locked Status**
	If set, all LFXO lockable registers are locked.			

Value	Mode	Description
0	UNLOCKED	LFXO lockable registers are not locked
1	LOCKED	LFXO lockable registers are locked

30:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	ENS	0x0	R	**LFXO Enable Status**
	LFXO is enabled.			
15:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	RDY	0x0	R	**LFXO Ready Status**
	LFXO is enabled and start-up time has exceeded.			

9.6.5.5 LFXO_CAL - LFXO Calibration Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x1			0x0						
Access																							RW			RW						
Name																							GAIN			CAPTUNE						

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	GAIN	0x1	RW	**LFXO Startup Gain**
	Optimal value depends on the chosen crystal. Please refer to the device data sheet for more information.			
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:0	CAPTUNE	0x0	RW	**Internal Capacitance Tuning**
	Program internal load capacitance connected between X_N pin and ground and X_P pin and ground. The bus affects tuning capacitances on both pins symmetrically. CAPTUNE value must not exceed 0x59. When updating CAPTUNE, its value must only be incremented or decremented by 1. Please refer to the device data sheet for more information.			

9.6.5.6 LFXO_IF - Interrupt Flag Register

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													FAIL	NEGEDGE	POSEDGE	RDY

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	FAIL	0x0	RW	**LFXO Failure Interrupt Flag**
	Set when LFXO failure is detected. Write 1 to clear the interrupt flag.			
2	NEGEDGE	0x0	RW	**Falling Edge Interrupt Flag**
	Triggers on every negative edge of the LFXO clock.			
1	POSEDGE	0x0	RW	**Rising Edge Interrupt Flag**
	Triggers on every positive edge of the LFXO clock.			
0	RDY	0x0	RW	**LFXO Ready Interrupt Flag**
	Set when LFXO is ready (start-up time exceeded). Write 1 to clear the interrupt flag.			

9.6.5.7 LFXO_IEN - Interrupt Enable Register

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													FAIL	NEGEDGE	POSEDGE	RDY

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	FAIL	0x0	RW	**LFXO Failure Interrupt Enable**
	Write 1 to enable FAILIF			
2	NEGEDGE	0x0	RW	**Falling Edge Interrupt Enable**
	Write 1 to enable NEGEDGEIF.			
1	POSEDGE	0x0	RW	**Rising Edge Interrupt Enable**
	Write 1 to enable POSEDGEIF.			
0	RDY	0x0	RW	**LFXO Ready Interrupt Enable**
	Write 1 to enable RDYIF.			

9.6.5.8 LFXO_SYNCBUSY - LFXO Sync Busy Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																CAL

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CAL	0x0	R	**LFXO Synchronization status**
	This bit is set when there is an ongoing synchronization of CAL register bitfields. Do not write to CAL register while this bit is set.			

9.6.5.9 LFXO_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x1A20															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x1A20	W	**Lock Key**
	Write any other value than UNLOCK to lock CTRL, CFG and CAL registers. Write UNLOCK value to unlock the registers.			

Value	Mode	Description
6688	UNLOCK	Unlock LFXO lockable registers

9.6.5.10 LFXO_AGCCTRL - AGC Control Register

Offset	Bit Position																														
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																DISAGCCLK

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	DISAGCCLK	0x0	RW	**LFXO AGC CLK Disable**
	Disable the clock to the digital agc			

Value	Mode	Description
0	ENABLED	ENABLE AGC CLK
1	DISABLED	DISABLE AGC CLK

9.7 LFRCO - Low-Frequency RC Oscillator

9.7.1 Introduction

The LFRCO is an integrated low-frequency (32.768 kHz) RC oscillator. It can be used as a timing reference in EM0, EM1, and EM4. A precision mode is available in EM0 and EM1. Precision mode enables hardware that periodically recalibrates the LFRCO against the HFXO crystal frequency when temperature changes to provide a fully internal 32.768 kHz clock source with +/-500 ppm accuracy.

9.7.2 Features

- 32.768 kHz oscillator
- High Accuracy
- Precision mode (+/-500 ppm) available in EM0 and EM1.
- Low-power non-precision operation available in EM0, EM1, and EM4.
- On-demand
- Lockable registers

9.7.3 Functional Description

9.7.3.1 Start-up

The LFRCO has a fast start-up time (refer to the data sheet electrical specifications for the exact start-up time). When the oscillator has started up and is ready to use, the RDY status bit will go high and the RDY interrupt will be triggered. After start-up, it may take two clock cycles for the clock to propagate through the CMU to the peripherals.

9.7.3.2 On-Demand Clocking

Software may forceably enable the LFRCO by setting the LFRCO_CTRL.FORCEEN bit field. However, by default, the LFRCO is configured to be enabled only when required by hardware, and to shut down when no hardware request is present (i.e. LFRCO_CTRL.DISONDEMAND=0 and LFRCO_CTRL.FORCEEN=0). This is known as on-demand clocking and allows the oscillator to be controlled without any software intervention.

9.7.3.3 Register Lock

The LFRCO configuration registers NOMCAL, NOMCALINV and CFG can be locked or unlocked by software using the LOCKKEY field in the LFRCO_LOCK register. By writing the UNLOCK value to LFRCO_LOCK.LOCKKEY, these registers will be unlocked and accessible to software. Any other value written to LFRCO_LOCK.LOCKKEY will lock the registers against write operations.

9.7.3.4 Precision Mode

Certain device families support a precision mode to bring the LFRCO accuracy to within +/-500 ppm, suitable for BLE sleep applications. Precision mode uses hardware to automatically re-calibrate the LFRCO against a crystal driven by the HFXO. Hardware detects temperature changes and initiates a re-calibration of the LFRCO as needed when operating in EM0 and EM1. If a re-calibration is necessary and the HFXO is not active, the precision mode hardware will automatically enable HFXO for a short time to perform the calibration. EM4 operation is not allowed while precision mode is enabled.

To enable precision mode software should set the LFRCO_CFG.HIGHPRECEN bit to 1 while the LFRCO is disabled (LFRCO_STATUS.ENS = 0). If this bit is written while the oscillator is enabled, a SYSCFG.TEALSYS interrupt will be generated. In a typical application, software will only access HIGHPRECEN at startup before any peripherals are configured to request the LFRCO.

Disabling precision mode is the inverse - software should ensure that no peripherals are requesting the LFRCO, and then clear the LFRCO_CFG.HIGHPRECEN bit to 0.

9.7.3.4.1 Reference Frequency

Precision mode uses a reference clock from the HFXO as a calibration target. Two registers (LFRCO_NOMCAL.NOMCALCNT and LFRCO_NOMCALINV.NOMCALCNTINV) are used to specify the nominal relationship between the HFXO reference clock and the LFRCO frequency as shown in Figure 9.2. LFRCO NOMCALCNT Calculation and Figure 9.3. LFRCO NOMCALCNTINV Calculation. NOMCALCNT and NOMCALCNTINV must be programmed while the LFRCO is not enabled.

$$NOMCALCNT = (320 * f_{HFXO}) / 32768$$

Where f_{HFXO} is the reference crystal frequency in Hz

Figure 9.2. LFRCO NOMCALCNT Calculation

$$NOMCALCNTINV = (1 / NOMCALCNT) * 2^{33}$$

Figure 9.3. LFRCO NOMCALCNTINV Calculation

9.7.3.4.2 Temperature Check and Calibration Intervals

When starting up in precision mode, the LFRCO will calibrate itself against the HFXO. After startup, the die temperature is checked periodically and if necessary, the LFRCO will initiate a re-calibration. Periodic re-calibration is also performed if the temperature has not changed for an extended period of time.

If a very high die temperature gradient is expected (e.g. due to high-power RF transmission), software can reduce the temperature check interval temporarily by setting the LFRCO_CMD.REDUCETCINT bit to 1. The LFRCO will then use a shorter temperature check interval until the chip enters a lower power sleep state. When sleep is entered, the temperature-check interval will gradually be increased based on the measured temperature gradient until it is back to the normal temperature check interval.

9.7.3.5 Interrupts

The LFRCO implements several interrupt flags in the LFRCO_IF register to report status, error events, or for debugging. Each interrupt flag has an enable bit in LFRCO_IEN. Setting a bit in IEN to 1 enables the corresponding interrupt source to trigger an LFRCO interrupt.

RDY is triggered after start-up, when the LFRCO startup sequence is complete and the oscillator is ready to use.

POSEDGE and NEGEDGE are triggered by the rising and falling edge of LFRCO respectively. These flags will only get set if either of the interrupts are enabled (with POSEDGE or NEGEDGE in the IEN register). Note that enabling NEGEDGE or POSEDGE act as a clock requester for the LFRCO oscillator, and these two interrupt enables must be disabled in order to disable the LFRCO.

RDY, POSEDGE, and NEGEDGE are available only in EM0 and EM1.

Three of the interrupt sources are error flags used by precision mode to alert software that there is a problem with the oscillator and it may no longer be running with full precision. CALOOR indicates that a calibration was performed, but the result was outside of the oscillator's adjustment range. TCOOR indicates that a temperature measurement was performed, but the measurement was out of range. Finally, the SCHEDERR flag indicates that a temperature check could not be performed due to a prior error. These three error flags are available when operating in precision mode in EM0 and EM1.

The remaining flags, TEMPCHANGE, CALDONE and TCDONE are available when operating in precision mode in EM0 and EM1 only and are provided for debugging purposes. Refer to the LFRCO_IF register description for the specific conditions of these flags.

9.7.4 LFRCO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LFRCO_IPVERSION	R	IP Version
0x004	LFRCO_CTRL	RW	Control Register
0x008	LFRCO_STATUS	RH	Status Register
0x014	LFRCO_IF	RWH INTFLAG	Interrupt Flag Register
0x018	LFRCO_IEN	RW	Interrupt Enable Register
0x020	LFRCO_LOCK	W	Configuration Lock Register
0x024	LFRCO_CFG	RW CONFIG	Configuration Register
0x02C	LFRCO_NOMCAL	RW CONFIG	Nominal Calibration Register
0x030	LFRCO_NOMCALINV	RW CONFIG	Nominal Calibration Inverted Register
0x034	LFRCO_CMD	W	Command Register
0x1000	LFRCO_IPVERSION_SET	R	IP Version
0x1004	LFRCO_CTRL_SET	RW	Control Register
0x1008	LFRCO_STATUS_SET	RH	Status Register
0x1014	LFRCO_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1018	LFRCO_IEN_SET	RW	Interrupt Enable Register
0x1020	LFRCO_LOCK_SET	W	Configuration Lock Register
0x1024	LFRCO_CFG_SET	RW CONFIG	Configuration Register
0x102C	LFRCO_NOMCAL_SET	RW CONFIG	Nominal Calibration Register
0x1030	LFRCO_NOMCALINV_SET	RW CONFIG	Nominal Calibration Inverted Register
0x1034	LFRCO_CMD_SET	W	Command Register
0x2000	LFRCO_IPVERSION_CLR	R	IP Version
0x2004	LFRCO_CTRL_CLR	RW	Control Register
0x2008	LFRCO_STATUS_CLR	RH	Status Register
0x2014	LFRCO_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2018	LFRCO_IEN_CLR	RW	Interrupt Enable Register
0x2020	LFRCO_LOCK_CLR	W	Configuration Lock Register
0x2024	LFRCO_CFG_CLR	RW CONFIG	Configuration Register
0x202C	LFRCO_NOMCAL_CLR	RW CONFIG	Nominal Calibration Register
0x2030	LFRCO_NOMCALINV_CLR	RW CONFIG	Nominal Calibration Inverted Register
0x2034	LFRCO_CMD_CLR	W	Command Register
0x3000	LFRCO_IPVERSION_TGL	R	IP Version
0x3004	LFRCO_CTRL_TGL	RW	Control Register
0x3008	LFRCO_STATUS_TGL	RH	Status Register
0x3014	LFRCO_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3018	LFRCO_IEN_TGL	RW	Interrupt Enable Register
0x3020	LFRCO_LOCK_TGL	W	Configuration Lock Register
0x3024	LFRCO_CFG_TGL	RW CONFIG	Configuration Register
0x302C	LFRCO_NOMCAL_TGL	RW CONFIG	Nominal Calibration Register

Offset	Name	Type	Description
0x3030	LFRCO_NOMCALINV_TGL	RW CONFIG	Nominal Calibration Inverted Register
0x3034	LFRCO_CMD_TGL	W	Command Register

9.7.5 LFRCO Register Description

9.7.5.1 LFRCO_IPVERSION - IP Version

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x4																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x4	R	**IP version ID**
	Reading this register returns the ip version number of LFRCO.			

9.7.5.2 LFRCO_CTRL - Control Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															DISONDEMAND	FORCEEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISONDEMAND	0x0	RW	**Disable On-Demand**
	Disable on demand functionality			
0	FORCEEN	0x0	RW	**Force Enable**
	Force the LFRCO core on			

9.7.5.3 LFRCO_STATUS - Status Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0															0x0																0x0
Access	R															R																R
Name	LOCK															ENS																RDY

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**Lock Status**
	This bit is set when LFRCO is locked.			

Value	Mode	Description
0	UNLOCKED	Access to configuration registers not locked
1	LOCKED	Access to configuration registers locked

30:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	ENS	0x0	R	**Enabled Status**
	This bit is set when LFRCO is enabling the analog core.			
15:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	RDY	0x0	R	**Ready Status**
	This bit is set when qualification is done and LFRCO is ready.			

9.7.5.4 LFRCO_IF - Interrupt Flag Register

Offset	Bit Position																														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset														0x0	0x0	0x0						0x0	0x0	0x0						0x0	0x0	0x0
Access														RW	RW	RW						RW	RW	RW						RW	RW	RW
Name														CALOOR	TCOOR	SCHEDERR						TEMPCHANGE	CALDONE	TCDONE						NEGEDGE	POSEDGE	RDY

Bit	Name	Reset	Access	Description
31:19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	CALOOR	0x0	RW	**Calibration Out Of Range Flag**
	Triggers if Calibration measure is out of range			
17	TCOOR	0x0	RW	**Temperature Check Out Of Range Flag**
	Triggers if Temperature Check measure is out of range			
16	SCHEDERR	0x0	RW	**Scheduling Error Flag**
	Triggers if a scheduled Temperature Check can not be handeled because prior Termperature Check or Calibration did not complete.			
15:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	TEMPCHANGE	0x0	RW	**Temperature Change Flag**
	Triggers when Temperature Check detects a change in temperature			
9	CALDONE	0x0	RW	**Calibration Done Flag**
	Triggers on completion of Calibration			
8	TCDONE	0x0	RW	**Temperature Check Done Flag**
	Triggers on completion of Temperature Check			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	NEGEDGE	0x0	RW	**Falling Edge Flag**
	Triggers on every negative edge of the LFRCO clock. IF will only be set when corresponding IEN is set.			
1	POSEDGE	0x0	RW	**Rising Edge Flag**
	Triggers on every positive edge of the LFRCO clock. IF will only be set when corresponding IEN is set.			
0	RDY	0x0	RW	**Ready Flag**
	Triggers when the oscillator becomes ready			

9.7.5.5 LFRCO_IEN - Interrupt Enable Register

Offset	Bit Position																														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset														0x0	0x0	0x0						0x0	0x0	0x0						0x0	0x0	0x0
Access														RW	RW	RW						RW	RW	RW						RW	RW	RW
Name														CALOOR	TCOOR	SCHEDERR						TEMPCHANGE	CALDONE	TCDONE						NEGEDGE	POSEDGE	RDY

Bit	Name	Reset	Access	Description
31:19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	CALOOR	0x0	RW	**Calibration Out Of Range Enable**
	Enables the Calibration Out Of Range interrupt			
17	TCOOR	0x0	RW	**Temperature Check Out Of Range Enable**
	Enables the Temperature Check Out Of Range interrupt			
16	SCHEDERR	0x0	RW	**Scheduling Error Enable**
	Enables the Scheduling Error interrupt			
15:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	TEMPCHANGE	0x0	RW	**Temperature Change Enable**
	Enables the Temperature Change interrupt			
9	CALDONE	0x0	RW	**Calibration Done Enable**
	Enables the Calibration Done interrupt			
8	TCDONE	0x0	RW	**Temperature Check Done Enable**
	Enables the Temperature Check Done interrupt			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	NEGEDGE	0x0	RW	**Falling Edge Enable**
	Enables the negedge interrupt and will cause the oscillator to run			
1	POSEDGE	0x0	RW	**Rising Edge Enable**
	Enables the posedge interrupt and will cause the oscillator to run			
0	RDY	0x0	RW	**Ready Enable**
	Enables the ready interrupt			

9.7.5.6 LFRCO_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x0	W	**Lock Key**

Writing the lock key will unlock the LFRCO configuration registers. Writing any other value will lock them.

Value	Mode	Description
0	LOCK	Lock Configuration Registers
3987	UNLOCK	Unlock Configuration Registers

9.7.5.7 LFRCO_CFG - Configuration Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																HIGHPRECEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	HIGHPRECEN	0x0	RW	**High Precision Enable**

LFRCO operates in High Precision Mode when this bit is set. HIGHPRECEN should not be written while LFRCO is enabled.

9.7.5.8 LFRCO_NOMCAL - Nominal Calibration Register

Offset	Bit Position																															
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset												0x5B8D8																				
Access												RW																				
Name												NOMCALCNT																				

Bit	Name	Reset	Access	Description
31:21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20:0	NOMCALCNT	0x5B8D8	RW	**Nominal Calibration Count** Expected Calibration count value. Should be set based on HFXO frequency; NOMCALCNT = 2 * f(HFXO) / (32.768 kHz / 160). Default value corresponds to 38.4 MHz HFXO frequency. NOMCALCNT should not be written while LFRCO is enabled.

9.7.5.9 LFRCO_NOMCALINV - Nominal Calibration Inverted Register

Offset	Bit Position																															
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x597A																
Access																RW																
Name																NOMCALCNTINV																

Bit	Name	Reset	Access	Description
31:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16:0	NOMCALCNTINV	0x597A	RW	**Nominal Calibration Count Inverted** This register should always be set to the inverse of NOMCALCNT value. Due to format of this register, integer value should be NOMCALCNTINV = (1 / NOMCALCNT) * (2**33). NOMCALCNTINV should not be written while LFRCO is enabled.

9.7.5.10 LFRCO_CMD - Command Register

Offset	Bit Position																															
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																W(nB)
Name																																REDUCETCINT

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	REDUCETCINT	0x0	W(nB)	**Reduce Temperature Check Interval**
	Setting this register field will temporarily lower the Temperature Check interval.			

9.8 FSRCO - Fast Start RCO

9.8.1 Introduction

This is an RC oscillator which can start and stop very fast. It is a fixed frequency oscillator, with no frequency configurability and as such any user of this clock can rely on it being a specific frequency independent of the system state. This is the first oscillator used during power up and hence it minimizes dependency to other blocks.

9.8.2 Features

- 20 MHz nominal frequency
- Low energy consumption

9.8.3 Functional Description

There are no programmable registers in this module. Software can choose to use this as system clock in the CMU block. the only way to enable or disable the FSRCO is by requesting it as a clock source in the CMU clock select registers.

9.8.4 FSRCO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	FSRCO_IPVERSION	R	IP Version
0x1000	FSRCO_IPVERSION_SET	R	IP Version
0x2000	FSRCO_IPVERSION_CLR	R	IP Version
0x3000	FSRCO_IPVERSION_TGL	R	IP Version

9.8.5 FSRCO Register Description

9.8.5.1 FSRCO_IPVERSION - IP Version

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x2																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x2	R	**IP Version**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

9.9 ULFRCO - Ultra Low Frequency RC Oscillator

9.9.1 Introduction

The ULFRCO is an ultra low power 1 kHz oscillator which is available in all energy modes. The ULFRCO is available to many low-frequency peripherals as a lower power alternative to one of the 32 kHz oscillators. This oscillator is also used for internal bias and housekeeping tasks.

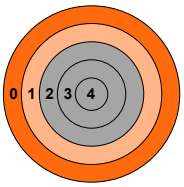
9.9.2 Features

- 1 kHz nominal frequency
- Low energy consumption

9.9.3 Functional Description

There are no user programmable registers in this module. The oscillator is always on in all energy modes except EM4. In EM4, the oscillator is available on-demand by peripheral requests.

10. SMU - Security Management Unit



Quick Facts

What?

The SMU provides configuration and status reporting for ARM TrustZone on the SiXG301.

Why?

Enables a robust solution at the system level.

How?

Hardware context switching and enhanced security provided by ARM TrustZone. Extension of the ARM MPU to control peripheral access.

10.1 Introduction

The Security Management Unit is used to configure and extend TrustZone bus level security provided by the Cortex-M33. In addition it increases the effective MPU regions by providing MPU control over peripheral access and memory regions.

10.2 Features

- Per peripheral privileged and secure attributes
- Per manager privileged and secure attributes
- Separate interrupt flags for privileged, secure, or instruction access exceptions
- Separate interrupt flag for secure manager access exceptions
- Secure and Privileged exception IRQs
- Configurable secure, non-secure, and non-secure-callable memory regions

10.3 Functional Description

10.3.1 Bus Level Security

Bus level security is the ability to control the flow of information on the device. The components of bus level security are the Cortex-M33, the Bus Manager Protection Unit (BMPU), the Peripheral Protection Unit (PPU), and the Memory Sector Protection Unit (MSPU) as highlighted in Figure 10.1. Bus Level Security Implementation. The SMU controls and configures all the components used in bus level security.

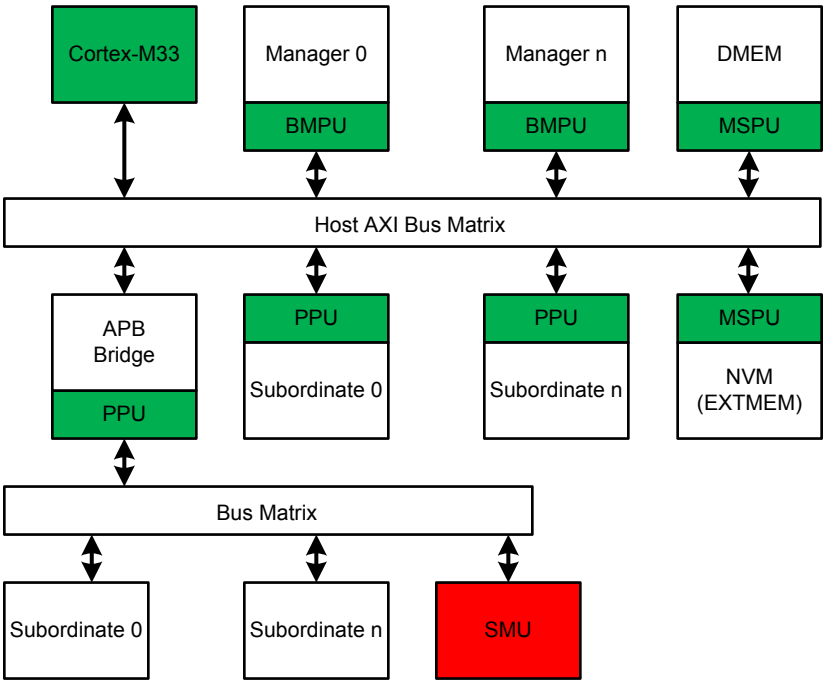


Figure 10.1. Bus Level Security Implementation

The BMPU is responsible for preventing managers (CPU, DMA, Etc..) from accessing secure addresses without authorization. For example, if a DMA configured as non-secure tries to access memory that is marked secure the BMPU will prevent access and set the corresponding interrupt flag. The BMPU prevents access of secure addresses by non-secure managers. The Cortex-M33 has BMPU functionality built into the TrustZone implementation.

The PPU is primarily responsible for blocking access to privileged peripherals from unprivileged managers. In addition, it also ensures that secure and non-secure peripherals are only accessible at the appropriate secure or non-secure addresses as described in 10.3.7 Configuring Peripherals in the PPU.

The MSPU assigns regions (i.e., sectors) of DMEM and NVM to secure states (Secure/Nonsecure). Attributed accesses to a given sector are checked against its access rights, and, if not allowed, the MSPU generates a fault response.

The Cortex-M33 also has an MPU which prevents execution of privileged memory when the CPU is in an unprivileged state. For more information on the MPU refer to the **ARM Cortex-M33 documentation**.

10.3.2 Secure and Non-Secure Register Blocks

The SMU provides two blocks of registers: SMU and SMU_CFGNS. The SMU register block is for the TrustZone secure machine, and SMU_CFGNS is for the TrustZone non-secure machine. Each register block has its own register lock. This allows for the priveledged assignments for both machines to be maintained in hardware. Documenation for this block focuses primarily on the secure machine configuration, but priveleged configuration within the non-secure machine is the same as the secure-machine with only the addition of a "NS" to differentiate the register files and registers. For example, PPU*, BMPU*, and LOCK registers in the secure machine are named PPUNS*, BMPUNS*, and NSLOCK in the non-secure machine. Note, the SMU_CFGNS register block is a subset of SMU, as security assignments occur only within secure machine.

10.3.3 Privileged Access Control

The Cortex-M33 and all other managers can be in either the privileged or unprivileged state. All bus access to peripherals are tested for privilege level by the PPU and resolved as shown in Table 10.1. Privileged Access Table.

Table 10.1. Privileged Access Table

Manager Attribute	Peripheral Attribute	Result
Privileged	Privileged	Success
Privileged	Unprivileged	Success
Unprivileged	Privileged	Exception
Unprivileged	Unprivileged	Success

10.3.4 Secure Access Control

The Cortex-M33 and all other managers can be in either the secure or non-secure state. All bus accesses are tested for security status by the BMPUs and PPUs and resolve as shown in Table 10.2. Secure Access Table. Secure access is computed using the secure attribute of the manager and the address region being accessed. If a peripheral is being accessed, the secure attribute of the peripheral is also used. For more information on the relationship between the address regions and peripheral security attributes please see 10.3.7 Configuring Peripherals in the PPU

Table 10.2. Secure Access Table

Manager Attribute	Address Attribute	Peripheral Attribute	Result
Secure	Secure	N/A	Success
Secure	Secure	Secure	Success
Secure	Secure	Non-secure	Success
Secure	Non-secure	N/A	Exception
Secure	Non-secure	Secure	Exception
Secure	Non-secure	Non-secure	Success
Non-secure	Secure	N/A	Exception
Non-secure	Secure	Secure	Exception
Non-secure	Secure	Non-secure	Exception
Non-secure	Non-secure	N/A	Success
Non-secure	Non-secure	Secure	Exception
Non-secure	Non-secure	Non-secure	Success

10.3.5 ARM TrustZone

ARM TrustZone is used to control what addresses are accessible by the CPU at any given time. There are two security states: secure and non-secure. In addition the MPU provides two privilege levels: privileged and unprivileged. This results in 4 possible states: secure-privileged, non-secure-privileged, secure-unprivileged and non-secure-unprivileged.

Non-secure code may not directly call secure code. To call secure code, non-secure code must first call a shim located in specially marked non-secure-callable memory. Unprivileged code may invoke privileged code and change the processor state to privileged by either issuing an SVC instruction or taking an interrupt. The processor is returned to unprivileged state when software manually reconfigures the security state or exits an interrupt.

For more information on secure/non-secure and privileged/unprivileged state transitions see the ARM Cortex-M33 documentation.

There are two primary use cases for TrustZone and the MPU. The first is simply partitioning a monolithic application in to the 4 states to protect some pieces of the system from bugs or attacks on others. The second is to use a RTOS to isolate several tasks from each other. In this case the RTOS itself normally consumes the privileged states with all other code running in the unprivileged states. Whenever a task switch occurs the RTOS can reconfigure the device so the new task has access to only the components it requires, protecting other tasks from interference.

In both use cases the TrustZone and MPU feature of the Cortex-M33 both secures and accelerates mode transitions while the SMU provides the ability to configure the security and privilege attributes of peripherals and memory.

The core is in secure-privileged state after a reset.

10.3.6 Configuring Managers in the BMPU

The SMU provides the ability to configure the current secure and privileged attribute of all bus managers except for the CPU via the BMPU (Bus Manager Protection Unit). The CPU is controlled as described in 10.3.5 ARM TrustZone.

To configure the privileged attribute of a manager set the appropriate bit in SMU_BMPUPATDn. To configure the secure attribute of a manager set the appropriate bit in SMU_BMPUSATDn.

10.3.7 Configuring Peripherals in the PPU

The SMU provides the ability to configure the current secure and privileged state of all peripherals via the PPU (Peripheral Protection Unit). To configure the privileged attribute of a peripheral set the appropriate bit in SMU_PPUPATDn.

Each peripheral is accessible at one of two addresses: A secure address and an non-secure address. Which address is valid depends on the security attribute of the peripheral configured in the SMU. When configured as secure a peripheral may only be accessed at its secure address and when configured as non-secure the peripheral may only be accessed at its non-secure address. This forces code to be aware of the security attribute of the peripheral being accessed, preventing secure code from accessing a non-secure peripheral unintentionally.

The device memory map contains 4 regions of fixed length and fixed security attribute to facilitate the secure access of peripherals and RF peripherals. There is one non-secure (0x40000000) and one secure (0x50000000) region for peripherals and one non-secure (0xA0000000) and secure (0xB0000000) region for the blocks in the radio subsystem.

To configure the security attribute of a peripheral set the appropriate bit in SMU_PPUSATDn.

10.3.8 Configuring ESAU Memory Regions

The SMU provides the ability to configure the security attribute of memory via the ESAU (External Security Attribute Unit). The ESAU implementation is based on ARM's recommendations, where all memory regions have a Secure and Non-Secure address alias. These are paired together, with the security level of each region determined by address bit 28. Secure regions are odd-numbered ESAU regions, where address bit 28 = 1. Non-secure regions are even-numbered ESAU regions, where address bit 28 = 0. Thus, ESAU region 0 and 1 are aliases, region 2 and 3 are aliases, and so on.

The ESAU includes an additional software programmable configuration to define the secure aliased region as either Secure or Secure/Non-Secure-Callable (S/NSC). Aliased regions that are designated non-secure are not impacted by the state of this bit, and remain as non-secure. The fields ESAURxNS in the ESAURTYPES register each cover a pair of aliased ESAU regions. The ESAURTYPES.ESAUEN bit enables the ESAU and applies the ESAURxNS attributes. When the ESAU is disabled, all configurable regions default to Secure.

Table 10.3. ESAU Memory Regions shows a common example where the Code secure region is configured as S/NSC, while the remaining secure regions are simply just Secure. Software may define Secure sub-regions in the code's S/NSC region, using the ARM's Security Attribution Unit (SAU).

Important: For devices with an ESAURTYPES.ESAUTZPOLARITY bit, this documentation assumes that ESAUTZPOLARITY bit is set to '1'. For device families without a ESAUTZPOLARITY bit, the ESAU allocations behave as though ESAUTZPOLARITY = 1. ESAUTZPOLARITY = 1 is the recommended setting for all implementations, but if user firmware chooses to set ESAUTZPOLARITY to '0', the meaning of bit 28 in the memory map is inverted.

Table 10.3. ESAU Memory Regions

Base Address	Limit Address	Type	ESAU Security	ESAU Region ID	ESAURTYPES S/NSC control bit
0xF0000000	0xFFFFFFFF	Device System	Exempt	15	--
0xE0000000	0xEFFFFFFF	Device System	Exempt	14	--
0xD0000000	0xDFFFFFFF	Device System	Secure	13	ESAUR12SNS
0xC0000000	0xCFFFFFFF	Device System	Non-Secure	12	
0xB0000000	0xBFFFFFFF	Device System	Secure	11	ESAUR10SNS
0xA0000000	0xAFFFFFFF	Device System	Non-Secure	10	
0x90000000	0x9FFFFFFF	Device RAM (WB)	Secure	9	ESAUR8SNS
0x80000000	0x8FFFFFFF	Device RAM (WB)	Non-Secure	8	
0x70000000	0x7FFFFFFF	Device RAM (WT)	Secure	7	ESAUR6SNS
0x60000000	0x6FFFFFFF	Device RAM (WT)	Non-Secure	6	
0x50000000	0x5FFFFFFF	Peripheral	Secure	5	ESAUR4SNS
0x40000000	0x4FFFFFFF	Peripheral	Non-Secure	4	
0x30000000	0x3FFFFFFF	SRAM	Secure	3	ESAUR2SNS
0x20000000	0x2FFFFFFF	SRAM	Non-Secure	2	
0x10000000	0x1FFFFFFF	Code	S/NSC	1	ESAUR0SNS
0x00000000	0x0FFFFFFF	Code	Non-Secure	0	

10.3.9 Configuring Memory in the MSPU

The SMU provides fine-grained security attribution within DMEM and NVM memory via the MSPU (Memory Sector Protection Unit). While the PPU assigns secure/non-secure attributes to each peripheral, the MSPU assigns secure/non-secure attributes to each granular block of DMEM and NVM. Secure/non-secure attributes for DMEM and NVM are configured independently by first setting the MSPURSIZE fields in MSPUDMEMREGIONSIZE and MSPUNVMREGIONSIZE to specify the size of each protection region within DMEM and NVM, respectively. Each can be divided into regions of the following sizes:

- 16 KBytes
- 32 KBytes
- 64 KBytes
- 128 KBytes
- 256 KBytes
- 512 KBytes

Given a specific region size, the subdivided memory regions are contiguous within the space, starting with region 0. For example, if MSPURSIZE is configured for 16 KBytes for DMEM, then Region 0 is address 0x00000000 to 0x00003FFF within that memory space, Region 1 is addresses 0x00004000 to 0x00007FFF, and so on.

Each block of this size within the memory is then attributed as being secure or nonsecure by setting the corresponding bit in the MSPUDMEMNSREGIONFLAGx and MSPUNVMNSREGIONFLAGx registers (one bit per region). These regions must then be accessed through the corresponding ESAU aperture corresponding to the security attributes of the initiating device. Sub-regions marked as secure can only be accessed by secure initiators acting through the secure ESAU address map, and regions marked as non-secure can only be accessed by non-secure initiators acting through the non-secure apertures of the ESAU address map.

10.3.10 Cortex-M33 Integration

In addition to the SMU based access controls the Cortex-M33 has additional security features for controlling both secure and privileged access.

The Security Attribution Unit (SAU) provides that ability to setup secure memory regions in addition to those configured by the SMU. To disable the SAU and rely entirely on the SMU for security management clear ENABLE and set ALLNS in the SAU CTRL register. To enable a combination of SMU and SAU control set ENABLE in the SAU CTRL register. If both ENABLE and ALLNS are cleared all Cortex-M33 will treat all transactions as secure.

When both SAU and SMU are in use, a memory address is considered secure if either the SAU or SMU have it configured as secure. When enabled the SAU applies ONLY to access by the Cortex-M33 and does not effect any other managers. For more information on the SAU refer to ARM documentation.

Note: It is highly recommended that systems avoid using only the SAU unless necessary. Since the SAU does not affect any managers outside the Cortex-M33, extreme care must be taken to ensure the SAU regions can not be trivially by bypassed through use of another manager such as the DMA.

In addition to the Cortex-M33 MPU provides the ability to control which regions of code and RAM are marked as privileged and prevent execution of privileged code by a CPU in unprivileged state. For more information on the configuration and use of the MPU refer to ARM documentation.

10.3.11 Exception Handling

When a fault is detected by the BMPU, PPU, or MSPU, the system can respond either with a bus error, or RAZ/WI (Read-as-Zero, Writes Ignored) and generate interrupts, along with information about the fault.

When any IRQ is trigged by the SMU the Cortex-M33 is automatically placed in the privileged state. The security state is determined by configuration inside the Cortex-M33. Refer to **ARM's documentation** for more details.

BMPU Faults

When a BMPU detects a non-secure manager attempting to access a secure address, the BMPUSEC in SMU_IF is set and the ID of the Manager block is written to SMU_BMPUFS. If SMU_IEN.BMPUSEC is set and the SMU's Secure IRQ enabled, the CPU will be interrupted.

The BMPURSPERRD0 register has individual bits to control the error response for each Manager. Bits in BMPURSPERRD0 should be set to '0' for RAZ/WI response, and '1' to generate a bus error.

PPU Faults

When a PPU detects an access to a secure peripheral at its non-secure address or an access to a non-secure peripheral at its secure address, PPUSEC in SMU_IF is set and the ID of the peripheral being accessed is written to SMU_PPUFS. If SMU_IEN.PPUSEC is set and the SMU's Secure IRQ is enabled, the CPU will be interrupted.

If a PPU detects an attempt to fetch an instruction from a peripheral, PPUINST in SMU_IF will be set and the ID of the peripheral being accessed is written to SMU_PPUFS. If SMU_IEN.PPUINST is set and the SMU's Privileged IRQ enabled, the CPU will be interrupted.

If a PPU detects an attempt to access a privileged peripheral by an unprivileged manager, PPUPRIV in SMU_IF will be set and the ID of the peripheral being accessed is written to SMU_PPUFS. If SMU_IEN.PPUPRIV is set and the SMU's Privileged IRQ enabled, the CPU will be interrupted.

The PPURSPERRD in the PPURSPERRD register controls the error response for PPU faults. PPURSPERRD should be set to '0' for RAZ/WI response, and '1' to generate a bus error.

MSPU Faults

The MSPU has separate fault detection for DMEM and NVM regions. If the MSPU detects an unallowed access, fault addresseses will be captured in the MSPUDMEMFSADDR or MSPUNVMFSADDR registers. MSPU faults accessing DMEM will set the SMU's MSPU0 interrupt flag, while NVM faults will set the MSPU1 interrupt flag in SMU_IF. If SMU_IEN.MSPUx is set and the SMU's Privileged IRQ enabled, the CPU will be interrupted.

The MSPURSPERRDx bits in the MSPUDMEMRSPERRD and MSPUNVMRSPERRD registers are used to control how the hardware will respond to an unallowed access. MSPURSPERRDx should be set to '0' for RAZ/WI response, and '1' to generate a bus error.

10.3.12 SMU Lock

The SMU registers can be locked to prevent unintended modifications. SMULOCK in SMU_STATUS indicates if the SMU is currently locked. To unlock the SMU write 0xACCE55 to the SMU_LOCK register. To lock write any other value to SMU_LOCK.

In addition to locking the SMU registers the SMU can prevent access to the Cortex-M33 ASU, MPU, SMPU, VTOR and VTAIRCR registers. To lock access to one or more of these blocks set the corresponding bit in SMU_M33CTRL.

10.4 SMU Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	SMU_IPVERSION	R	IP Version
0x004	SMU_STATUS	RH	Status Register
0x008	SMU_LOCK	W	Lock Register
0x00C	SMU_IF	RWH INTFLAG	Interrupt Flag Register
0x010	SMU_IEN	RW	Interrupt Enable Register
0x020	SMU_M33CTRL	RW	M33 Control Settings
0x040	SMU_PPUPATD0	RW	PPU Privileged Access
0x044	SMU_PPUPATD1	RW	PPU Privileged Access
0x060	SMU_PPUSATD0	RW	PPU Secure Access
0x064	SMU_PPUSATD1	RW	PPU Secure Access
0x130	SMU_PPURSPERRD	RW	PPU Respond With Error
0x140	SMU_PPUFS	RH	PPU Fault Status
0x144	SMU_PPUFSADDR	RH	PPU Fault Status Address
0x150	SMU_BMPUPATD0	RW	BMPU Privileged Attribute
0x170	SMU_BMPUSATD0	RW	BMPU Secure Attribute
0x190	SMU_BMPURSPERRD0	RW	BMPU Respond With Error
0x250	SMU_BMPUFS	RH	BMPU Fault Status
0x254	SMU_BMPUFSADDR	RH	BMPU Fault Status Address
0x260	SMU_MSPUDMEMREGIONSIZE	RWH	DMEM MSPU Region Size
0x264	SMU_MSPUNVMREGIONSIZE	RWH	NVM MSPU Region Size
0x268	SMU_MSPUDMEMRSPERRD	RW	DMEM MSPU Respond With Error
0x26C	SMU_MSPUNVMRSPERRD	RW	NVM MSPU Respond With Error
0x270	SMU_MSPUDMEMNSREGION-FLAG0	RW	DMEM MSPU Non-Secure Region Flags
0x274	SMU_MSPUNVMNSREGION-FLAG0	RW	NVM MSPU Non-Secure Region Flags
0x278	SMU_MSPUNVMNSREGION-FLAG1	RW	NVM MSPU Non-Secure Region Flags
0x27C	SMU_MSPUNVMNSREGION-FLAG2	RW	NVM MSPU Non-Secure Region Flags
0x280	SMU_MSPUNVMNSREGION-FLAG3	RW	NVM MSPU Non-Secure Region Flags
0x360	SMU_MSPUDMEMFSADDR	RH	DMEM MSPU Fault Status Address
0x364	SMU_MSPUNVMFSADDR	RH	NVM MSPU Fault Status Address
0x370	SMU_ESAURTYPES	RW	ESAU Region Types
0x1000	SMU_IPVERSION_SET	R	IP Version
0x1004	SMU_STATUS_SET	RH	Status Register
0x1008	SMU_LOCK_SET	W	Lock Register
0x100C	SMU_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1010	SMU_IEN_SET	RW	Interrupt Enable Register
0x1020	SMU_M33CTRL_SET	RW	M33 Control Settings

Offset	Name	Type	Description
0x1040	SMU_PPUPATD0_SET	RW	PPU Privileged Access
0x1044	SMU_PPUPATD1_SET	RW	PPU Privileged Access
0x1060	SMU_PPUSATD0_SET	RW	PPU Secure Access
0x1064	SMU_PPUSATD1_SET	RW	PPU Secure Access
0x1130	SMU_PPURSPERRD_SET	RW	PPU Respond With Error
0x1140	SMU_PPUFS_SET	RH	PPU Fault Status
0x1144	SMU_PPUFSADDR_SET	RH	PPU Fault Status Address
0x1150	SMU_BMPUPATD0_SET	RW	BMPU Privileged Attribute
0x1170	SMU_BMPUSATD0_SET	RW	BMPU Secure Attribute
0x1190	SMU_BMPURSPERRD0_SET	RW	BMPU Respond With Error
0x1250	SMU_BMPUFS_SET	RH	BMPU Fault Status
0x1254	SMU_BMPUFSADDR_SET	RH	BMPU Fault Status Address
0x1260	SMU_MSPUDMEMREGION-SIZE_SET	RWH	DMEM MSPU Region Size
0x1264	SMU_MSPUNVMREGION-SIZE_SET	RWH	NVM MSPU Region Size
0x1268	SMU_MSPUDMEMR-SPERRD_SET	RW	DMEM MSPU Respond With Error
0x126C	SMU_MSPUNVMR-SPERRD_SET	RW	NVM MSPU Respond With Error
0x1270	SMU_MSPUDMEMNSREGION-FLAG0_SET	RW	DMEM MSPU Non-Secure Region Flags
0x1274	SMU_MSPUNVMNSREGION-FLAG0_SET	RW	NVM MSPU Non-Secure Region Flags
0x1278	SMU_MSPUNVMNSREGION-FLAG1_SET	RW	NVM MSPU Non-Secure Region Flags
0x127C	SMU_MSPUNVMNSREGION-FLAG2_SET	RW	NVM MSPU Non-Secure Region Flags
0x1280	SMU_MSPUNVMNSREGION-FLAG3_SET	RW	NVM MSPU Non-Secure Region Flags
0x1360	SMU_MSPUDMEMF-SADDR_SET	RH	DMEM MSPU Fault Status Address
0x1364	SMU_MSPUNVMFSADDR_SET	RH	NVM MSPU Fault Status Address
0x1370	SMU_ESAURTYPES_SET	RW	ESAU Region Types
0x2000	SMU_IPVERSION_CLR	R	IP Version
0x2004	SMU_STATUS_CLR	RH	Status Register
0x2008	SMU_LOCK_CLR	W	Lock Register
0x200C	SMU_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2010	SMU_IEN_CLR	RW	Interrupt Enable Register
0x2020	SMU_M33CTRL_CLR	RW	M33 Control Settings
0x2040	SMU_PPUPATD0_CLR	RW	PPU Privileged Access
0x2044	SMU_PPUPATD1_CLR	RW	PPU Privileged Access
0x2060	SMU_PPUSATD0_CLR	RW	PPU Secure Access
0x2064	SMU_PPUSATD1_CLR	RW	PPU Secure Access
0x2130	SMU_PPURSPERRD_CLR	RW	PPU Respond With Error

Offset	Name	Type	Description
0x2140	SMU_PPUFS_CLR	RH	PPU Fault Status
0x2144	SMU_PPUFSADDR_CLR	RH	PPU Fault Status Address
0x2150	SMU_BMPUPATD0_CLR	RW	BMPU Privileged Attribute
0x2170	SMU_BMPUSATD0_CLR	RW	BMPU Secure Attribute
0x2190	SMU_BMPURSPERRD0_CLR	RW	BMPU Respond With Error
0x2250	SMU_BMPUFS_CLR	RH	BMPU Fault Status
0x2254	SMU_BMPUFSADDR_CLR	RH	BMPU Fault Status Address
0x2260	SMU_MSPUDMEMREGION-SIZE_CLR	RWH	DMEM MSPU Region Size
0x2264	SMU_MSPUNVMREGION-SIZE_CLR	RWH	NVM MSPU Region Size
0x2268	SMU_MSPUDMEMR-SPERRD_CLR	RW	DMEM MSPU Respond With Error
0x226C	SMU_MSPUNVMR-SPERRD_CLR	RW	NVM MSPU Respond With Error
0x2270	SMU_MSPUDMEMNSREGION-FLAG0_CLR	RW	DMEM MSPU Non-Secure Region Flags
0x2274	SMU_MSPUNVMNSREGION-FLAG0_CLR	RW	NVM MSPU Non-Secure Region Flags
0x2278	SMU_MSPUNVMNSREGION-FLAG1_CLR	RW	NVM MSPU Non-Secure Region Flags
0x227C	SMU_MSPUNVMNSREGION-FLAG2_CLR	RW	NVM MSPU Non-Secure Region Flags
0x2280	SMU_MSPUNVMNSREGION-FLAG3_CLR	RW	NVM MSPU Non-Secure Region Flags
0x2360	SMU_MSPUDMEMF-SADDR_CLR	RH	DMEM MSPU Fault Status Address
0x2364	SMU_MSPUNVMFSADDR_CLR	RH	NVM MSPU Fault Status Address
0x2370	SMU_ESAURTYPES_CLR	RW	ESAU Region Types
0x3000	SMU_IPVERSION_TGL	R	IP Version
0x3004	SMU_STATUS_TGL	RH	Status Register
0x3008	SMU_LOCK_TGL	W	Lock Register
0x300C	SMU_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3010	SMU_IEN_TGL	RW	Interrupt Enable Register
0x3020	SMU_M33CTRL_TGL	RW	M33 Control Settings
0x3040	SMU_PPUPATD0_TGL	RW	PPU Privileged Access
0x3044	SMU_PPUPATD1_TGL	RW	PPU Privileged Access
0x3060	SMU_PPUSATD0_TGL	RW	PPU Secure Access
0x3064	SMU_PPUSATD1_TGL	RW	PPU Secure Access
0x3130	SMU_PPURSPERRD_TGL	RW	PPU Respond With Error
0x3140	SMU_PPUFS_TGL	RH	PPU Fault Status
0x3144	SMU_PPUFSADDR_TGL	RH	PPU Fault Status Address
0x3150	SMU_BMPUPATD0_TGL	RW	BMPU Privileged Attribute
0x3170	SMU_BMPUSATD0_TGL	RW	BMPU Secure Attribute
0x3190	SMU_BMPURSPERRD0_TGL	RW	BMPU Respond With Error

Offset	Name	Type	Description
0x3250	SMU_BMPUFS_TGL	RH	BMPU Fault Status
0x3254	SMU_BMPUFSADDR_TGL	RH	BMPU Fault Status Address
0x3260	SMU_MSPUDMEMREGION-SIZE_TGL	RWH	DMEM MSPU Region Size
0x3264	SMU_MSPUNVMREGION-SIZE_TGL	RWH	NVM MSPU Region Size
0x3268	SMU_MSPUDMEMR-SPERRD_TGL	RW	DMEM MSPU Respond With Error
0x326C	SMU_MSPUNVMR-SPERRD_TGL	RW	NVM MSPU Respond With Error
0x3270	SMU_MSPUDMEMNSREGION-FLAG0_TGL	RW	DMEM MSPU Non-Secure Region Flags
0x3274	SMU_MSPUNVMNSREGION-FLAG0_TGL	RW	NVM MSPU Non-Secure Region Flags
0x3278	SMU_MSPUNVMNSREGION-FLAG1_TGL	RW	NVM MSPU Non-Secure Region Flags
0x327C	SMU_MSPUNVMNSREGION-FLAG2_TGL	RW	NVM MSPU Non-Secure Region Flags
0x3280	SMU_MSPUNVMNSREGION-FLAG3_TGL	RW	NVM MSPU Non-Secure Region Flags
0x3360	SMU_MSPUDMEMF-SADDR_TGL	RH	DMEM MSPU Fault Status Address
0x3364	SMU_MSPUNVMFSADDR_TGL	RH	NVM MSPU Fault Status Address
0x3370	SMU_ESAURTYPES_TGL	RW	ESAU Region Types

10.5 SMU Register Description

10.5.1 SMU_IPVERSION - IP Version

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x8																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x8	R	**IP Version**
	Gives access to the Revision ID of SMU			

10.5.2 SMU_STATUS - Status Register

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																SMULOCK

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	SMULOCK	0x0	R	**SMU Lock**

Indicates if SMU Registers are locked.

Value	Mode	Description
0	UNLOCKED	
1	LOCKED	

10.5.3 SMU_LOCK - Lock Register

Offset	Bit Position																															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																							
Access									W																							
Name									SMULOCKKEY																							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	SMULOCKKEY	0x0	W	

Write anything but UNLOCK to lock registers.

Value	Mode	Description
11325013	UNLOCK	Unlocks Registers

10.5.4 SMU_IF - Interrupt Flag Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	0x0										0x0		0x0		0x0		0x0
Access															RW	RW										RW		RW		RW		RW
Name															BMPUSEC	PPUSEC										MSPU1		MSPU0		PPUINST		PPUPRIV

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	BMPUSEC	0x0	RW	**BMPU Security Interrupt Flag**
	Triggered when a security fault occurs in the Bus Master Protection Unit			
16	PPUSEC	0x0	RW	**PPU Security Interrupt Flag**
	Triggered when a security fault occurs in the Peripheral Protection Unit			
15:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	MSPU1	0x0	RW	**NVM MSPU Interrupt Flag**
	Triggered when a status fault occurs in the NVM MSPU Unit			
5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	MSPU0	0x0	RW	**DMEM MSPU Interrupt Flag**
	Triggered when a status fault occurs in the DMEM MSPU Unit			
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	PPUINST	0x0	RW	**PPU Instruction Interrupt Flag**
	Triggered when a instruction fault occurs in the Peripheral Protection Unit			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	PPUPRIV	0x0	RW	**PPU Privilege Interrupt Flag**
	Triggered when a privilege fault occurs in the Peripheral Protection Unit			

10.5.5 SMU_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	0x0										0x0		0x0		0x0		0x0
Access															RW	RW										RW		RW		RW		RW
Name															BMPUSEC	PPUSEC										MSPU1		MSPU0		PPUINST		PPUPRIV

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	BMPUSEC	0x0	RW	**BMPU Security Interrupt Enable**
	Set to enable the BMPUSEC Interrupt			
16	PPUSEC	0x0	RW	**PPU Security Interrupt Enable**
	Set to enable the PPUSEC Interrupt			
15:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	MSPU1	0x0	RW	**NVM MSPU Interrupt Enable**
	Set to enable the NVM MSPU1 Interrupt			
5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	MSPU0	0x0	RW	**DMEM MSPU Interrupt Enable**
	Set to enable the DMEM MSPU0 Interrupt			
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	PPUINST	0x0	RW	**PPU Instruction Interrupt Enable**
	Set to enable the PPUINST Interrupt			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	PPUPRIV	0x0	RW	**PPU Privilege Interrupt Enable**
	Set to enable the PPUPRIV Interrupt			

10.5.6 SMU_M33CTRL - M33 Control Settings

Offset	Bit Position																														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												LOCKSAU	LOCKNSMPU	LOCKSMPU	LOCKNSVTOR	LOCKSVTAIRCR

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	LOCKSAU	0x0	RW	**Lock SAU**
	Set to 1 to lock security attribution unit			
3	LOCKNSMPU	0x0	RW	**Lock Non-secure MPU configuration**
	Set to 1 to lock non-secure MPU configuration			
2	LOCKSMPU	0x0	RW	**Lock Secure MPU configuration**
	Set to 1 to lock secure MPU configuration			
1	LOCKNSVTOR	0x0	RW	**Lock Non-secure VTOR**
	Set to 1 to lock non-secure VTOR			
0	LOCKSVTAIRCR	0x0	RW	**Lock Secure VTAIRCR**
	Set to 1 to lock secure VTAIRCR			

10.5.7 SMU_PPUPATD0 - PPU Privileged Access

Offset	Bit Position																															
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name	EUSART1	LEDDRV0	GPCRC0	BURAM	SYSCFG	SYSCFGCFGNS	CHIPTESTCTRL	I2C1	BURTC	PIXELRZ0	L2ICACHE0	ETAMPDET	TIMER3	TIMER2	TIMER1	TIMER0	LDMAXBAR0	GPIO	PRS	L1ICACHE0	SEMAPHORE1	SEMAPHORE0	ULFRCO	LFRCO	LFXO	DPLL0	FSRCO	LDMA0	HFRCO0	DMEM	CMU	SOCPLL0

Bit	Name	Reset	Access	Description
31	EUSART1	0x1	RW	**EUSART1 Privileged Access**
	EUSART1 Privileged Access			
30	LEDDRV0	0x1	RW	**LEDDRV0 Privileged Access**
	LEDDRV0 Privileged Access			
29	GPCRC0	0x1	RW	**GPCRC0 Privileged Access**
	GPCRC0 Privileged Access			
28	BURAM	0x1	RW	**BURAM Privileged Access**
	BURAM Privileged Access			
27	SYSCFG	0x1	RW	**SYSCFG Privileged Access**
	SYSCFG Privileged Access			
26	SYSCFGCFGNS	0x1	RW	**SYSCFGCFGNS Privileged Access**
	SYSCFGCFGNS Privileged Access			
25	CHIPTESTCTRL	0x1	RW	**CHIPTESTCTRL Privileged Access**
	CHIPTESTCTRL Privileged Access			
24	I2C1	0x1	RW	**I2C1 Privileged Access**
	I2C1 Privileged Access			
23	BURTC	0x1	RW	**BURTC Privileged Access**
	BURTC Privileged Access			
22	PIXELRZ0	0x1	RW	**PIXELRZ0 Privileged Access**
	PIXELRZ0 Privileged Access			
21	L2ICACHE0	0x1	RW	**L2ICACHE0 Privileged Access**
	L2ICACHE0 Privileged Access			
20	ETAMPDET	0x1	RW	**ETAMPDET Privileged Access**
	ETAMPDET Privileged Access			
19	TIMER3	0x1	RW	**TIMER3 Privileged Access**
	TIMER3 Privileged Access			
18	TIMER2	0x1	RW	**TIMER2 Privileged Access**
	TIMER2 Privileged Access			
17	TIMER1	0x1	RW	**TIMER1 Privileged Access**
	TIMER1 Privileged Access			

Bit	Name	Reset	Access	Description
16	TIMER0	0x1	RW	**TIMER0 Privileged Access**
	TIMER0 Privileged Access			
15	LDMAXBAR0	0x1	RW	**LDMAXBAR0 Privileged Access**
	LDMAXBAR0 Privileged Access			
14	GPIO	0x1	RW	**GPIO Privileged Access**
	GPIO Privileged Access			
13	PRS	0x1	RW	**PRS Privileged Access**
	PRS Privileged Access			
12	L1ICACHE0	0x1	RW	**L1ICACHE0 Privileged Access**
	L1ICACHE0 Privileged Access			
11	SEMAPHORE1	0x1	RW	**SEMAPHORE1 Privileged Access**
	SEMAPHORE1 Privileged Access			
10	SEMAPHORE0	0x1	RW	**SEMAPHORE0 Privileged Access**
	SEMAPHORE0 Privileged Access			
9	ULFRCO	0x1	RW	**ULFRCO Privileged Access**
	ULFRCO Privileged Access			
8	LFRCO	0x1	RW	**LFRCO Privileged Access**
	LFRCO Privileged Access			
7	LFXO	0x1	RW	**LFXO Privileged Access**
	LFXO Privileged Access			
6	DPLL0	0x1	RW	**DPLL0 Privileged Access**
	DPLL0 Privileged Access			
5	FSRCO	0x1	RW	**FSRCO Privileged Access**
	FSRCO Privileged Access			
4	LDMA0	0x1	RW	**LDMA0 Privileged Access**
	LDMA0 Privileged Access			
3	HFRCO0	0x1	RW	**HFRCO0 Privileged Access**
	HFRCO0 Privileged Access			
2	DMEM	0x1	RW	**DMEM Privileged Access**
	DMEM Privileged Access			
1	CMU	0x1	RW	**CMU Privileged Access**
	CMU Privileged Access			
0	SOCPLL0	0x1	RW	**SOCPLL0 Privileged Access**
	SOCPLL0 Privileged Access			

10.5.8 SMU_PPUPATD1 - PPU Privileged Access

Offset	Bit Position																															
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name	HOSTPORTAL	LPW0PORTAL	SEPORTAL	EMUCFGNS	EMU	SCRATCHPAD	SYSRTC0	AHBRADIO	QSPI0MANUAL	SEMAILBOX	SMUCFGNS	SMU	LPWAESDMACTRL	SYMCRYPTODMACTRL	EUSART0	WDOG0	I2C0	HFXO0	HFRCOEM23	PCNT0	AMUXCP0	ACMP1	ACMP0	ADC0	LETIMER0	DEVINFO	KSU	RPA	WDOG1	I2C2	PIXELRZ1	EUSART2

Bit	Name	Reset	Access	Description
31	HOSTPORTAL	0x1	RW	**HOSTPORTAL Privileged Access**
	HOSTPORTAL Privileged Access			
30	LPW0PORTAL	0x1	RW	**LPW0PORTAL Privileged Access**
	LPW0PORTAL Privileged Access			
29	SEPORTAL	0x1	RW	**SEPORTAL Privileged Access**
	SEPORTAL Privileged Access			
28	EMUCFGNS	0x1	RW	**EMUCFGNS Privileged Access**
	EMUCFGNS Privileged Access			
27	EMU	0x1	RW	**EMU Privileged Access**
	EMU Privileged Access			
26	SCRATCHPAD	0x1	RW	**SCRATCHPAD Privileged Access**
	SCRATCHPAD Privileged Access			
25	SYSRTC0	0x1	RW	**SYSRTC0 Privileged Access**
	SYSRTC0 Privileged Access			
24	AHBRADIO	0x1	RW	**AHBRADIO Privileged Access**
	AHBRADIO Privileged Access			
23	QSPI0MANUAL	0x1	RW	**QSPI0MANUAL Privileged Access**
	QSPI0MANUAL Privileged Access			
22	SEMAILBOX	0x1	RW	**SEMAILBOX Privileged Access**
	SEMAILBOX Privileged Access			
21	SMUCFGNS	0x1	RW	**SMUCFGNS Privileged Access**
	SMUCFGNS Privileged Access			
20	SMU	0x1	RW	**SMU Privileged Access**
	SMU Privileged Access			
19	LPWAESDMACTRL	0x1	RW	**LPWAESDMACTRL Privileged Access**
	LPWAESDMACTRL Privileged Access			
18	SYMCRYPTOD-MACTRL	0x1	RW	**SYMCRYPTODMACTRL Privileged Access**
	SYMCRYPTODMACTRL Privileged Access			

Bit	Name	Reset	Access	Description
17	EUSART0	0x1	RW	**EUSART0 Privileged Access**
	EUSART0 Privileged Access			
16	WDOG0	0x1	RW	**WDOG0 Privileged Access**
	WDOG0 Privileged Access			
15	I2C0	0x1	RW	**I2C0 Privileged Access**
	I2C0 Privileged Access			
14	HFXO0	0x1	RW	**HFXO0 Privileged Access**
	HFXO0 Privileged Access			
13	HFRCOEM23	0x1	RW	**HFRCOEM23 Privileged Access**
	HFRCOEM23 Privileged Access			
12	PCNT0	0x1	RW	**PCNT0 Privileged Access**
	PCNT0 Privileged Access			
11	AMUXCP0	0x1	RW	**AMUXCP0 Privileged Access**
	AMUXCP0 Privileged Access			
10	ACMP1	0x1	RW	**ACMP1 Privileged Access**
	ACMP1 Privileged Access			
9	ACMP0	0x1	RW	**ACMP0 Privileged Access**
	ACMP0 Privileged Access			
8	ADC0	0x1	RW	**ADC0 Privileged Access**
	ADC0 Privileged Access			
7	LETIMER0	0x1	RW	**LETIMER0 Privileged Access**
	LETIMER0 Privileged Access			
6	DEVINFO	0x1	RW	**DEVINFO Privileged Access**
	DEVINFO Privileged Access			
5	KSU	0x1	RW	**KSU Privileged Access**
	KSU Privileged Access			
4	RPA	0x1	RW	**RPA Privileged Access**
	RPA Privileged Access			
3	WDOG1	0x1	RW	**WDOG1 Privileged Access**
	WDOG1 Privileged Access			
2	I2C2	0x1	RW	**I2C2 Privileged Access**
	I2C2 Privileged Access			
1	PIXELRZ1	0x1	RW	**PIXELRZ1 Privileged Access**
	PIXELRZ1 Privileged Access			
0	EUSART2	0x1	RW	**EUSART2 Privileged Access**
	EUSART2 Privileged Access			

10.5.9 SMU_PPUSATD0 - PPU Secure Access

Offset	Bit Position																															
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name	EUSART1	LEDDRV0	GPCRC0	BURAM	SYSCFG	SYSCFGCFGNS	CHIPTESTCTRL	I2C1	BURTC	PIXELRZ0	L2ICACHE0	ETAMPDET	TIMER3	TIMER2	TIMER1	TIMER0	LDMAXBAR0	GPIO	PRS	L1ICACHE0	SEMAPHORE1	SEMAPHORE0	ULFRCO	LFRCO	LFXO	DPLL0	FSRCO	LDMA0	HFRCO0	DMEM	CMU	SOCPLL0

Bit	Name	Reset	Access	Description
31	EUSART1	0x1	RW	**EUSART1 Secure Access**
	EUSART1 Secure Access			
30	LEDDRV0	0x1	RW	**LEDDRV0 Secure Access**
	LEDDRV0 Secure Access			
29	GPCRC0	0x1	RW	**GPCRC0 Secure Access**
	GPCRC0 Secure Access			
28	BURAM	0x1	RW	**BURAM Secure Access**
	BURAM Secure Access			
27	SYSCFG	0x1	RW	**SYSCFG Secure Access**
	SYSCFG Secure Access			
26	SYSCFGCFGNS	0x1	RW	**SYSCFGCFGNS Secure Access**
	SYSCFGCFGNS Secure Access			
25	CHIPTESTCTRL	0x1	RW	**CHIPTESTCTRL Secure Access**
	CHIPTESTCTRL Secure Access			
24	I2C1	0x1	RW	**I2C1 Secure Access**
	I2C1 Secure Access			
23	BURTC	0x1	RW	**BURTC Secure Access**
	BURTC Secure Access			
22	PIXELRZ0	0x1	RW	**PIXELRZ0 Secure Access**
	PIXELRZ0 Secure Access			
21	L2ICACHE0	0x1	RW	**L2ICACHE0 Secure Access**
	L2ICACHE0 Secure Access			
20	ETAMPDET	0x1	RW	**ETAMPDET Secure Access**
	ETAMPDET Secure Access			
19	TIMER3	0x1	RW	**TIMER3 Secure Access**
	TIMER3 Secure Access			
18	TIMER2	0x1	RW	**TIMER2 Secure Access**
	TIMER2 Secure Access			
17	TIMER1	0x1	RW	**TIMER1 Secure Access**
	TIMER1 Secure Access			

Bit	Name	Reset	Access	Description
16	TIMER0	0x1	RW	**TIMER0 Secure Access**
	TIMER0 Secure Access			
15	LDMAXBAR0	0x1	RW	**LDMAXBAR0 Secure Access**
	LDMAXBAR0 Secure Access			
14	GPIO	0x1	RW	**GPIO Secure Access**
	GPIO Secure Access			
13	PRS	0x1	RW	**PRS Secure Access**
	PRS Secure Access			
12	L1ICACHE0	0x1	RW	**L1ICACHE0 Secure Access**
	L1ICACHE0 Secure Access			
11	SEMAPHORE1	0x1	RW	**SEMAPHORE1 Secure Access**
	SEMAPHORE1 Secure Access			
10	SEMAPHORE0	0x1	RW	**SEMAPHORE0 Secure Access**
	SEMAPHORE0 Secure Access			
9	ULFRCO	0x1	RW	**ULFRCO Secure Access**
	ULFRCO Secure Access			
8	LFRCO	0x1	RW	**LFRCO Secure Access**
	LFRCO Secure Access			
7	LFXO	0x1	RW	**LFXO Secure Access**
	LFXO Secure Access			
6	DPLL0	0x1	RW	**DPLL0 Secure Access**
	DPLL0 Secure Access			
5	FSRCO	0x1	RW	**FSRCO Secure Access**
	FSRCO Secure Access			
4	LDMA0	0x1	RW	**LDMA0 Secure Access**
	LDMA0 Secure Access			
3	HFRCO0	0x1	RW	**HFRCO0 Secure Access**
	HFRCO0 Secure Access			
2	DMEM	0x1	RW	**DMEM Secure Access**
	DMEM Secure Access			
1	CMU	0x1	RW	**CMU Secure Access**
	CMU Secure Access			
0	SOCPLL0	0x1	RW	**SOCPLL0 Secure Access**
	SOCPLL0 Secure Access			

10.5.10 SMU_PPUSATD1 - PPU Secure Access

Offset	Bit Position																															
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name	HOSTPORTAL	LPW0PORTAL	SEPORTAL	EMUCFGNS	EMU	SCRATCHPAD	SYSRTC0	AHBRADIO	QSPI0MANUAL	SEMAILBOX	SMUCFGNS	SMU	LPWAESDMACTRL	SYMCRYPTODMACTRL	EUSART0	WDOG0	I2C0	HFXO0	HFRCOEM23	PCNT0	AMUXCP0	ACMP1	ACMP0	ADC0	LETIMER0	DEVINFO	KSU	RPA	WDOG1	I2C2	PIXELRZ1	EUSART2

Bit	Name	Reset	Access	Description
31	HOSTPORTAL	0x1	RW	**HOSTPORTAL Secure Access**
	HOSTPORTAL Secure Access			
30	LPW0PORTAL	0x1	RW	**LPW0PORTAL Secure Access**
	LPW0PORTAL Secure Access			
29	SEPORTAL	0x1	RW	**SEPORTAL Secure Access**
	SEPORTAL Secure Access			
28	EMUCFGNS	0x1	RW	**EMUCFGNS Secure Access**
	EMUCFGNS Secure Access			
27	EMU	0x1	RW	**EMU Secure Access**
	EMU Secure Access			
26	SCRATCHPAD	0x1	RW	**SCRATCHPAD Secure Access**
	SCRATCHPAD Secure Access			
25	SYSRTC0	0x1	RW	**SYSRTC0 Secure Access**
	SYSRTC0 Secure Access			
24	AHBRADIO	0x1	RW	**AHBRADIO Secure Access**
	AHBRADIO Secure Access			
23	QSPI0MANUAL	0x1	RW	**QSPI0MANUAL Secure Access**
	QSPI0MANUAL Secure Access			
22	SEMAILBOX	0x1	RW	**SEMAILBOX Secure Access**
	SEMAILBOX Secure Access			
21	SMUCFGNS	0x1	RW	**SMUCFGNS Secure Access**
	SMUCFGNS Secure Access			
20	SMU	0x1	RW	**SMU Secure Access**
	SMU Secure Access			
19	LPWAESDMACTRL	0x1	RW	**LPWAESDMACTRL Secure Access**
	LPWAESDMACTRL Secure Access			
18	SYMCRYPTOD-MACTRL	0x1	RW	**SYMCRYPTODMACTRL Secure Access**
	SYMCRYPTODMACTRL Secure Access			

Bit	Name	Reset	Access	Description
17	EUSART0 EUSART0 Secure Access	0x1	RW	**EUSART0 Secure Access**
16	WDOG0 WDOG0 Secure Access	0x1	RW	**WDOG0 Secure Access**
15	I2C0 I2C0 Secure Access	0x1	RW	**I2C0 Secure Access**
14	HFXO0 HFXO0 Secure Access	0x1	RW	**HFXO0 Secure Access**
13	HFRCOEM23 HFRCOEM23 Secure Access	0x1	RW	**HFRCOEM23 Secure Access**
12	PCNT0 PCNT0 Secure Access	0x1	RW	**PCNT0 Secure Access**
11	AMUXCP0 AMUXCP0 Secure Access	0x1	RW	**AMUXCP0 Secure Access**
10	ACMP1 ACMP1 Secure Access	0x1	RW	**ACMP1 Secure Access**
9	ACMP0 ACMP0 Secure Access	0x1	RW	**ACMP0 Secure Access**
8	ADC0 ADC0 Secure Access	0x1	RW	**ADC0 Secure Access**
7	LETIMER0 LETIMER0 Secure Access	0x1	RW	**LETIMER0 Secure Access**
6	DEVINFO DEVINFO Secure Access	0x1	RW	**DEVINFO Secure Access**
5	KSU KSU Secure Access	0x1	RW	**KSU Secure Access**
4	RPA RPA Secure Access	0x1	RW	**RPA Secure Access**
3	WDOG1 WDOG1 Secure Access	0x1	RW	**WDOG1 Secure Access**
2	I2C2 I2C2 Secure Access	0x1	RW	**I2C2 Secure Access**
1	PIXELRZ1 PIXELRZ1 Secure Access	0x1	RW	**PIXELRZ1 Secure Access**
0	EUSART2 EUSART2 Secure Access	0x1	RW	**EUSART2 Secure Access**

10.5.11 SMU_PPURSPERRD - PPU Respond With Error

Offset	Bit Position																													
0x130	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																PPURSPERRD

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	PPURSPERRD	0x0	RW	**PPU Respond with Error**
	Set to 0 to respond with RAZ/WI, and 1 to respond with a bus error			

10.5.12 SMU_PPUFS - PPU Fault Status

Offset	Bit Position																													
0x140	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									PPUFSPERIPHID							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	PPUFSPERIPHID	0x0	R	**Peripheral ID**
	ID of the peripheral that caused the fault.			

10.5.13 SMU_PPUFSADDR - PPU Fault Status Address

Offset	Bit Position																															
0x144	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	PPUFSADDR																															

Bit	Name	Reset	Access	Description
31:0	PPUFSADDR	0x0	R	**Fault Address**
	Access address that triggered fault			

10.5.14 SMU_BMPUPATD0 - BMPU Privileged Attribute

Offset	Bit Position																													
0x150	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
Access																									RW	RW	RW	RW	RW	RW	RW	RW
Name																									SLAXICOREBUS	SYMCRYPTODMA	SEEXTDMAPORT	RPASLAXI	LDMA0PORT1	LDMA0PORT0	AHBRADIOBUS	LPWAESDMA

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	SLAXICOREBUS	0x1	RW	**SLAXICOREBUS Privileged Mode**
	SLAXICOREBUS Privileged Mode			
6	SYMCRYPTODMA	0x1	RW	**SYMCRYPTODMA Privileged Mode**
	SYMCRYPTODMA Privileged Mode			
5	SEEXTDMAPORT	0x1	RW	**SEEXTDMAPORT Privileged Mode**
	SEEXTDMAPORT Privileged Mode			
4	RPASLAXI	0x1	RW	**RPASLAXI Privileged Mode**
	RPASLAXI Privileged Mode			
3	LDMA0PORT1	0x1	RW	**LDMA0PORT1 Privileged Mode**
	LDMA0PORT1 Privileged Mode			
2	LDMA0PORT0	0x1	RW	**LDMA0PORT0 Privileged Mode**
	LDMA0PORT0 Privileged Mode			
1	AHBRADIOBUS	0x1	RW	**AHBRADIOBUS Privileged Mode**
	AHBRADIOBUS Privileged Mode			
0	LPWAESDMA	0x1	RW	**LPWAESDMA Privileged Mode**
	LPWAESDMA Privileged Mode			

10.5.15 SMU_BMPUSATD0 - BMPU Secure Attribute

Offset	Bit Position																													
0x170	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
Access																									RW	RW	RW	RW	RW	RW	RW	RW
Name																									SLAXICOREBUS	SYMCRYPTODMA	SEEXTDMAPORT	RPASLAXI	LDMA0PORT1	LDMA0PORT0	AHBRADIOBUS	LPWAESDMA

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	SLAXICOREBUS SLAXICOREBUS Secure Mode	0x1	RW	**SLAXICOREBUS Secure Mode**
6	SYMCRYPTODMA SYMCRYPTODMA Secure Mode	0x1	RW	**SYMCRYPTODMA Secure Mode**
5	SEEXTDMAPORT SEEXTDMAPORT Secure Mode	0x1	RW	**SEEXTDMAPORT Secure Mode**
4	RPASLAXI RPASLAXI Secure Mode	0x1	RW	**RPASLAXI Secure Mode**
3	LDMA0PORT1 LDMA0PORT1 Secure Mode	0x1	RW	**LDMA0PORT1 Secure Mode**
2	LDMA0PORT0 LDMA0PORT0 Secure Mode	0x1	RW	**LDMA0PORT0 Secure Mode**
1	AHBRADIOBUS AHBRADIOBUS Secure Mode	0x1	RW	**AHBRADIOBUS Secure Mode**
0	LPWAESDMA LPWAESDMA Secure Mode	0x1	RW	**LPWAESDMA Secure Mode**

10.5.16 SMU_BMPURSPERRD0 - BMPU Respond With Error

Offset	Bit Position																													
0x190	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																									RW	RW	RW	RW	RW	RW	RW	RW
Name																									SLAXICOREBUS	SYMCRYPTODMA	SEEXTDMAPORT	RPASLAXI	LDMA0PORT1	LDMA0PORT0	AHBRADIOBUS	LPWAESDMA

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	SLAXICOREBUS	0x0	RW	**SLAXICOREBUS Respond with Error**
	SLAXICOREBUS Respond with Error			
6	SYMCRYPTODMA	0x0	RW	**SYMCRYPTODMA Respond with Error**
	SYMCRYPTODMA Respond with Error			
5	SEEXTDMAPORT	0x0	RW	**SEEXTDMAPORT Respond with Error**
	SEEXTDMAPORT Respond with Error			
4	RPASLAXI	0x0	RW	**RPASLAXI Respond with Error**
	RPASLAXI Respond with Error			
3	LDMA0PORT1	0x0	RW	**LDMA0PORT1 Respond with Error**
	LDMA0PORT1 Respond with Error			
2	LDMA0PORT0	0x0	RW	**LDMA0PORT0 Respond with Error**
	LDMA0PORT0 Respond with Error			
1	AHBRADIOBUS	0x0	RW	**AHBRADIOBUS Respond with Error**
	AHBRADIOBUS Respond with Error			
0	LPWAESDMA	0x0	RW	**LPWAESDMA Respond with Error**
	LPWAESDMA Respond with Error			

10.5.17 SMU_BMPUFS - BMPU Fault Status

Offset	Bit Position																													
0x250	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									BMPUFSMASTERID							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	BMPUFSMASTERID	0x0	R	**Bus Manager ID**
	ID of bus manager that triggered fault			

10.5.18 SMU_BMPUFSADDR - BMPU Fault Status Address

Offset	Bit Position																													
0x254	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	BMPUFSADDR																															

Bit	Name	Reset	Access	Description
31:0	BMPUFSADDR	0x0	R	**Fault Address**
	Access address that triggered fault			

10.5.19 SMU_MSPUDMEMREGIONSIZE - DMEM MSPU Region Size

Offset	Bit Position																														
0x260	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0		
Access																														RW		
Name																														MSPURSIZE		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	MSPURSIZE	0x0	RW	**Region Size configuration**

Region Size is the size of each Region of DMEM MSPU

Value	Mode	Description
0	MSPU_16KB	
1	MSPU_32KB	
2	MSPU_64KB	
3	MSPU_128KB	
4	MSPU_256KB	
5	MSPU_512KB	

10.5.20 SMU_MSPUNVMREGIONSIZE - NVM MSPU Region Size

Offset	Bit Position																														
0x264	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0		
Access																														RW		
Name																														MSPURSIZE		

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	MSPURSIZE	0x0	RW	**Region Size configuration**
	Region Size is the size of each Region of NVM MSPU			

Value	Mode	Description
0	MSPU_16KB	
1	MSPU_32KB	
2	MSPU_64KB	
3	MSPU_128KB	
4	MSPU_256KB	
5	MSPU_512KB	

10.5.21 SMU_MSPUDMEMRSPERRD - DMEM MSPU Respond With Error

Offset	Bit Position																														
0x268	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x1
Access																																RW
Name																																MSPURSPERRD0

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	MSPURSPERRD0	0x1	RW	**DMEM MSPU Respond with Error**
	Set to 0 to respond with RAZ/WI, and 1 to respond with a bus error			

10.5.22 SMU_MSPUNVMRSPERRD - NVM MSPU Respond With Error

Offset	Bit Position																														
0x26C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x1
Access																																RW
Name																																MSPURSPERRD0

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	MSPURSPERRD0	0x1	RW	**NVM MSPU Respond with Error** Set to 0 to respond with RAZ/WI, and 1 to respond with a bus error

10.5.23 SMU_MSPUDMEMNSREGIONFLAG0 - DMEM MSPU Non-Secure Region Flags

Offset	Bit Position																															
0x270	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name	MSPUNSRFL31	MSPUNSRFL30	MSPUNSRFL29	MSPUNSRFL28	MSPUNSRFL27	MSPUNSRFL26	MSPUNSRFL25	MSPUNSRFL24	MSPUNSRFL23	MSPUNSRFL22	MSPUNSRFL21	MSPUNSRFL20	MSPUNSRFL19	MSPUNSRFL18	MSPUNSRFL17	MSPUNSRFL16	MSPUNSRFL15	MSPUNSRFL14	MSPUNSRFL13	MSPUNSRFL12	MSPUNSRFL11	MSPUNSRFL10	MSPUNSRFL9	MSPUNSRFL8	MSPUNSRFL7	MSPUNSRFL6	MSPUNSRFL5	MSPUNSRFL4	MSPUNSRFL3	MSPUNSRFL2	MSPUNSRFL1	MSPUNSRFL0

Bit	Name	Reset	Access	Description
31	MSPUNSRFL31	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
30	MSPUNSRFL30	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
29	MSPUNSRFL29	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
28	MSPUNSRFL28	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
27	MSPUNSRFL27	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
26	MSPUNSRFL26	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
25	MSPUNSRFL25	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
24	MSPUNSRFL24	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
23	MSPUNSRFL23	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
22	MSPUNSRFL22	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
21	MSPUNSRFL21	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
20	MSPUNSRFL20	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
19	MSPUNSRFL19	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
18	MSPUNSRFL18	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
17	MSPUNSRFL17	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
16	MSPUNSRFL16	0x0	RW	**DMEM non-secure region flags**

Bit	Name	Reset	Access	Description
	Set to 0 for secure region and set to 1 for non-secure region			
15	MSPUNSRFL15	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
14	MSPUNSRFL14	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
13	MSPUNSRFL13	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
12	MSPUNSRFL12	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
11	MSPUNSRFL11	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
10	MSPUNSRFL10	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
9	MSPUNSRFL9	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
8	MSPUNSRFL8	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
7	MSPUNSRFL7	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
6	MSPUNSRFL6	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
5	MSPUNSRFL5	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
4	MSPUNSRFL4	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
3	MSPUNSRFL3	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
2	MSPUNSRFL2	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
1	MSPUNSRFL1	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
0	MSPUNSRFL0	0x0	RW	**DMEM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			

10.5.24 SMU_MSPUNVMNSREGIONFLAG0 - NVM MSPU Non-Secure Region Flags

Offset	Bit Position																															
0x274	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name	MSPUNSRFL31_0	MSPUNSRFL30_0	MSPUNSRFL29_0	MSPUNSRFL28_0	MSPUNSRFL27_0	MSPUNSRFL26_0	MSPUNSRFL25_0	MSPUNSRFL24_0	MSPUNSRFL23_0	MSPUNSRFL22_0	MSPUNSRFL21_0	MSPUNSRFL20_0	MSPUNSRFL19_0	MSPUNSRFL18_0	MSPUNSRFL17_0	MSPUNSRFL16_0	MSPUNSRFL15_0	MSPUNSRFL14_0	MSPUNSRFL13_0	MSPUNSRFL12_0	MSPUNSRFL11_0	MSPUNSRFL10_0	MSPUNSRFL9_0	MSPUNSRFL8_0	MSPUNSRFL7_0	MSPUNSRFL6_0	MSPUNSRFL5_0	MSPUNSRFL4_0	MSPUNSRFL3_0	MSPUNSRFL2_0	MSPUNSRFL1_0	MSPUNSRFL0_0

Bit	Name	Reset	Access	Description
31	MSPUNSRFL31_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
30	MSPUNSRFL30_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
29	MSPUNSRFL29_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
28	MSPUNSRFL28_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
27	MSPUNSRFL27_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
26	MSPUNSRFL26_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
25	MSPUNSRFL25_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
24	MSPUNSRFL24_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
23	MSPUNSRFL23_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
22	MSPUNSRFL22_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
21	MSPUNSRFL21_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
20	MSPUNSRFL20_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
19	MSPUNSRFL19_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
18	MSPUNSRFL18_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
17	MSPUNSRFL17_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			

Bit	Name	Reset	Access	Description
16	MSPUNSRFL16_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
15	MSPUNSRFL15_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
14	MSPUNSRFL14_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
13	MSPUNSRFL13_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
12	MSPUNSRFL12_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
11	MSPUNSRFL11_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
10	MSPUNSRFL10_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
9	MSPUNSRFL9_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
8	MSPUNSRFL8_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
7	MSPUNSRFL7_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
6	MSPUNSRFL6_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
5	MSPUNSRFL5_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
4	MSPUNSRFL4_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
3	MSPUNSRFL3_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
2	MSPUNSRFL2_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
1	MSPUNSRFL1_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
0	MSPUNSRFL0_0	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			

10.5.25 SMU_MSPUNVMNSREGIONFLAG1 - NVM MSPU Non-Secure Region Flags

Offset	Bit Position																															
0x278	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name	MSPUNSRFL31_1	MSPUNSRFL30_1	MSPUNSRFL29_1	MSPUNSRFL28_1	MSPUNSRFL27_1	MSPUNSRFL26_1	MSPUNSRFL25_1	MSPUNSRFL24_1	MSPUNSRFL23_1	MSPUNSRFL22_1	MSPUNSRFL21_1	MSPUNSRFL20_1	MSPUNSRFL19_1	MSPUNSRFL18_1	MSPUNSRFL17_1	MSPUNSRFL16_1	MSPUNSRFL15_1	MSPUNSRFL14_1	MSPUNSRFL13_1	MSPUNSRFL12_1	MSPUNSRFL11_1	MSPUNSRFL10_1	MSPUNSRFL9_1	MSPUNSRFL8_1	MSPUNSRFL7_1	MSPUNSRFL6_1	MSPUNSRFL5_1	MSPUNSRFL4_1	MSPUNSRFL3_1	MSPUNSRFL2_1	MSPUNSRFL1_1	MSPUNSRFL0_1

Bit	Name	Reset	Access	Description
31	MSPUNSRFL31_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
30	MSPUNSRFL30_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
29	MSPUNSRFL29_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
28	MSPUNSRFL28_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
27	MSPUNSRFL27_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
26	MSPUNSRFL26_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
25	MSPUNSRFL25_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
24	MSPUNSRFL24_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
23	MSPUNSRFL23_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
22	MSPUNSRFL22_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
21	MSPUNSRFL21_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
20	MSPUNSRFL20_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
19	MSPUNSRFL19_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
18	MSPUNSRFL18_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
17	MSPUNSRFL17_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			

Bit	Name	Reset	Access	Description
16	MSPUNSRFL16_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
15	MSPUNSRFL15_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
14	MSPUNSRFL14_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
13	MSPUNSRFL13_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
12	MSPUNSRFL12_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
11	MSPUNSRFL11_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
10	MSPUNSRFL10_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
9	MSPUNSRFL9_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
8	MSPUNSRFL8_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
7	MSPUNSRFL7_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
6	MSPUNSRFL6_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
5	MSPUNSRFL5_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
4	MSPUNSRFL4_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
3	MSPUNSRFL3_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
2	MSPUNSRFL2_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
1	MSPUNSRFL1_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
0	MSPUNSRFL0_1	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			

10.5.26 SMU_MSPUNVMNSREGIONFLAG2 - NVM MSPU Non-Secure Region Flags

Offset	Bit Position																															
0x27C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name	MSPUNSRFL31_2	MSPUNSRFL30_2	MSPUNSRFL29_2	MSPUNSRFL28_2	MSPUNSRFL27_2	MSPUNSRFL26_2	MSPUNSRFL25_2	MSPUNSRFL24_2	MSPUNSRFL23_2	MSPUNSRFL22_2	MSPUNSRFL21_2	MSPUNSRFL20_2	MSPUNSRFL19_2	MSPUNSRFL18_2	MSPUNSRFL17_2	MSPUNSRFL16_2	MSPUNSRFL15_2	MSPUNSRFL14_2	MSPUNSRFL13_2	MSPUNSRFL12_2	MSPUNSRFL11_2	MSPUNSRFL10_2	MSPUNSRFL9_2	MSPUNSRFL8_2	MSPUNSRFL7_2	MSPUNSRFL6_2	MSPUNSRFL5_2	MSPUNSRFL4_2	MSPUNSRFL3_2	MSPUNSRFL2_2	MSPUNSRFL1_2	MSPUNSRFL0_2

Bit	Name	Reset	Access	Description
31	MSPUNSRFL31_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
30	MSPUNSRFL30_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
29	MSPUNSRFL29_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
28	MSPUNSRFL28_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
27	MSPUNSRFL27_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
26	MSPUNSRFL26_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
25	MSPUNSRFL25_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
24	MSPUNSRFL24_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
23	MSPUNSRFL23_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
22	MSPUNSRFL22_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
21	MSPUNSRFL21_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
20	MSPUNSRFL20_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
19	MSPUNSRFL19_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
18	MSPUNSRFL18_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
17	MSPUNSRFL17_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			

Bit	Name	Reset	Access	Description
16	MSPUNSRFL16_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
15	MSPUNSRFL15_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
14	MSPUNSRFL14_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
13	MSPUNSRFL13_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
12	MSPUNSRFL12_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
11	MSPUNSRFL11_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
10	MSPUNSRFL10_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
9	MSPUNSRFL9_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
8	MSPUNSRFL8_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
7	MSPUNSRFL7_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
6	MSPUNSRFL6_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
5	MSPUNSRFL5_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
4	MSPUNSRFL4_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
3	MSPUNSRFL3_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
2	MSPUNSRFL2_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
1	MSPUNSRFL1_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
0	MSPUNSRFL0_2	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			

10.5.27 SMU_MSPUNVMNSREGIONFLAG3 - NVM MSPU Non-Secure Region Flags

Offset	Bit Position																															
0x280	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name	MSPUNSRFL31_3	MSPUNSRFL30_3	MSPUNSRFL29_3	MSPUNSRFL28_3	MSPUNSRFL27_3	MSPUNSRFL26_3	MSPUNSRFL25_3	MSPUNSRFL24_3	MSPUNSRFL23_3	MSPUNSRFL22_3	MSPUNSRFL21_3	MSPUNSRFL20_3	MSPUNSRFL19_3	MSPUNSRFL18_3	MSPUNSRFL17_3	MSPUNSRFL16_3	MSPUNSRFL15_3	MSPUNSRFL14_3	MSPUNSRFL13_3	MSPUNSRFL12_3	MSPUNSRFL11_3	MSPUNSRFL10_3	MSPUNSRFL9_3	MSPUNSRFL8_3	MSPUNSRFL7_3	MSPUNSRFL6_3	MSPUNSRFL5_3	MSPUNSRFL4_3	MSPUNSRFL3_3	MSPUNSRFL2_3	MSPUNSRFL1_3	MSPUNSRFL0_3

Bit	Name	Reset	Access	Description
31	MSPUNSRFL31_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
30	MSPUNSRFL30_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
29	MSPUNSRFL29_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
28	MSPUNSRFL28_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
27	MSPUNSRFL27_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
26	MSPUNSRFL26_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
25	MSPUNSRFL25_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
24	MSPUNSRFL24_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
23	MSPUNSRFL23_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
22	MSPUNSRFL22_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
21	MSPUNSRFL21_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
20	MSPUNSRFL20_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
19	MSPUNSRFL19_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
18	MSPUNSRFL18_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
17	MSPUNSRFL17_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			

Bit	Name	Reset	Access	Description
16	MSPUNSRFL16_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
15	MSPUNSRFL15_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
14	MSPUNSRFL14_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
13	MSPUNSRFL13_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
12	MSPUNSRFL12_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
11	MSPUNSRFL11_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
10	MSPUNSRFL10_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
9	MSPUNSRFL9_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
8	MSPUNSRFL8_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
7	MSPUNSRFL7_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
6	MSPUNSRFL6_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
5	MSPUNSRFL5_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
4	MSPUNSRFL4_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
3	MSPUNSRFL3_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
2	MSPUNSRFL2_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
1	MSPUNSRFL1_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			
0	MSPUNSRFL0_3	0x0	RW	**NVM non-secure region flags**
	Set to 0 for secure region and set to 1 for non-secure region			

10.5.28 SMU_MSPUDMEMFSADDR - DMEM MSPU Fault Status Address

Offset	Bit Position																													
0x360	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	MSPUFSADDR																															

Bit	Name	Reset	Access	Description
31:0	MSPUFSADDR	0x0	R	**Fault Address**
	Access address that triggered fault			

10.5.29 SMU_MSPUNVMFSADDR - NVM MSPU Fault Status Address

Offset	Bit Position																													
0x364	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	MSPUFSADDR																															

Bit	Name	Reset	Access	Description
31:0	MSPUFSADDR	0x0	R	**Fault Address**
	Access address that triggered fault			

10.5.30 SMU_ESAURTYPES - ESAU Region Types

Offset	Bit Position																													
0x370	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																		0x0		0x0		0x0		0x0		0x0		0x0		0x0		0x0
Access																		RW		RW		RW		RW		RW		RW		RW		RW
Name																		ESAUR12SNS		ESAUR10SNS		ESAUR8SNS		ESAUR6SNS		ESAUR4SNS		ESAUR2SNS		ESAUR0SNS		ESAUEN

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14	ESAUR12SNS	0x0	RW	**Region 12/13 Non-Secure-Callable**
	Set to 1 to configure Region 12 or 13 as Non-secure-Callable			
13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	ESAUR10SNS	0x0	RW	**Region 10/11 Non-Secure-Callable**
	Set to 1 to configure Region 10 or 11 as Non-secure-Callable			
11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	ESAUR8SNS	0x0	RW	**Region 8/9 Non-Secure-Callable**
	Set to 1 to configure Region 8 or 9 as Non-secure-Callable			
9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	ESAUR6SNS	0x0	RW	**Region 6/7 Non-Secure-Callable**
	Set to 1 to configure Region 6 or 7 as Non-secure-Callable			
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	ESAUR4SNS	0x0	RW	**Region 4/5 Non-Secure-Callable**
	Set to 1 to configure Region 4 or 5 as Non-secure-Callable			
5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	ESAUR2SNS	0x0	RW	**Region 2/3 Non-Secure-Callable**
	Set to 1 to configure Region 2 or 3 as Non-secure-Callable			
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	ESAUR0SNS	0x0	RW	**Region 0/1 Non-Secure-Callable**
	Set to 1 to configure Region 0 or 1 as Non-secure-Callable			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	ESAUEN	0x0	RW	**ESAU enable**
	ESAUEN determines if the ESAU is enabled or disabled, and it is disabled by default (all regions are marked as Secure)			

10.6 SMU_CFGNS Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x004	SMU_NSSTATUS	RH	Status Register
0x008	SMU_NSLOCK	W	Lock Register
0x00C	SMU_NSIF	RWH INTFLAG	Interrupt Flag Register
0x010	SMU_NSIEN	RW	Interrupt Enable Register
0x040	SMU_PPUNSPATD0	RW	PPU Privileged Access
0x044	SMU_PPUNSPATD1	RW	PPU Privileged Access
0x140	SMU_PPUNSFS	RH	PPU Fault Status
0x144	SMU_PPUNSFSADDR	RH	PPU Fault Status Address
0x150	SMU_BMPUNSPATD0	RW	BMPU Privileged Attribute
0x1004	SMU_NSSTATUS_SET	RH	Status Register
0x1008	SMU_NSLOCK_SET	W	Lock Register
0x100C	SMU_NSIF_SET	RWH INTFLAG	Interrupt Flag Register
0x1010	SMU_NSIEN_SET	RW	Interrupt Enable Register
0x1040	SMU_PPUNSPATD0_SET	RW	PPU Privileged Access
0x1044	SMU_PPUNSPATD1_SET	RW	PPU Privileged Access
0x1140	SMU_PPUNSFS_SET	RH	PPU Fault Status
0x1144	SMU_PPUNSFSADDR_SET	RH	PPU Fault Status Address
0x1150	SMU_BMPUNSPATD0_SET	RW	BMPU Privileged Attribute
0x2004	SMU_NSSTATUS_CLR	RH	Status Register
0x2008	SMU_NSLOCK_CLR	W	Lock Register
0x200C	SMU_NSIF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2010	SMU_NSIEN_CLR	RW	Interrupt Enable Register
0x2040	SMU_PPUNSPATD0_CLR	RW	PPU Privileged Access
0x2044	SMU_PPUNSPATD1_CLR	RW	PPU Privileged Access
0x2140	SMU_PPUNSFS_CLR	RH	PPU Fault Status
0x2144	SMU_PPUNSFSADDR_CLR	RH	PPU Fault Status Address
0x2150	SMU_BMPUNSPATD0_CLR	RW	BMPU Privileged Attribute
0x3004	SMU_NSSTATUS_TGL	RH	Status Register
0x3008	SMU_NSLOCK_TGL	W	Lock Register
0x300C	SMU_NSIF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3010	SMU_NSIEN_TGL	RW	Interrupt Enable Register
0x3040	SMU_PPUNSPATD0_TGL	RW	PPU Privileged Access
0x3044	SMU_PPUNSPATD1_TGL	RW	PPU Privileged Access
0x3140	SMU_PPUNSFS_TGL	RH	PPU Fault Status
0x3144	SMU_PPUNSFSADDR_TGL	RH	PPU Fault Status Address
0x3150	SMU_BMPUNSPATD0_TGL	RW	BMPU Privileged Attribute

10.7 SMU_CFGNS Register Description

10.7.1 SMU_NSSTATUS - Status Register

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																SMUNSLOCK

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	SMUNSLOCK	0x0	R	**SMUNS Lock**
	Indicates if SMUNS Registers are locked.			

Value	Mode	Description
0	UNLOCKED	
1	LOCKED	

10.7.2 SMU_NSLOCK - Lock Register

Offset	Bit Position																															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																							
Access									W																							
Name									SMUNSLOCKKEY																							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	SMUNSLOCKKEY	0x0	W	
	Write anything but UNLOCK to lock registers.			

Value	Mode	Description
11325013	UNLOCK	Unlocks Registers

10.7.3 SMU_NSIF - Interrupt Flag Register

Offset	Bit Position																														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0		0x0
Access																														RW		RW
Name																														PPUNSINST		PPUNSPRIV

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	PPUNSINST	0x0	RW	**PPUNS Instruction Interrupt Flag**
	Triggered when an instruction fault occurs in the Peripheral Protection Unit			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	PPUNSPRIV	0x0	RW	**PPUNS Privilege Interrupt Flag**
	Triggered when a privilege fault occurs in the Peripheral Protection Unit			

10.7.4 SMU_NSIEN - Interrupt Enable Register

Offset	Bit Position																														
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0		0x0
Access																														RW		RW
Name																														PPUNSINST		PPUNSPRIV

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	PPUNSINST	0x0	RW	**PPUNS Instruction Interrupt Enable**
	Set to enable the PPUNSINST Interrupt			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	PPUNSPRIV	0x0	RW	**PPUNS Privilege Interrupt Enable**
	Set to enable the PPUNSPRI Interrupt			

10.7.5 SMU_PPUNSPATD0 - PPU Privileged Access

Offset	Bit Position																															
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name	EUSART1	LEDDRV0	GPCRC0	BURAM	SYSCFG	SYSCFGCFGNS	CHIPTESTCTRL	I2C1	BURTC	PIXELRZ0	L2ICACHE0	ETAMPDET	TIMER3	TIMER2	TIMER1	TIMER0	LDMAXBAR0	GPIO	PRS	L1ICACHE0	SEMAPHORE1	SEMAPHORE0	ULFRCO	LFRCO	LFXO	DPLL0	FSRCO	LDMA0	HFRCO0	DMEM	CMU	SOCPLL0

Bit	Name	Reset	Access	Description
31	EUSART1	0x1	RW	**EUSART1 Privileged Access**
	EUSART1 Privileged Access			
30	LEDDRV0	0x1	RW	**LEDDRV0 Privileged Access**
	LEDDRV0 Privileged Access			
29	GPCRC0	0x1	RW	**GPCRC0 Privileged Access**
	GPCRC0 Privileged Access			
28	BURAM	0x1	RW	**BURAM Privileged Access**
	BURAM Privileged Access			
27	SYSCFG	0x1	RW	**SYSCFG Privileged Access**
	SYSCFG Privileged Access			
26	SYSCFGCFGNS	0x1	RW	**SYSCFGCFGNS Privileged Access**
	SYSCFGCFGNS Privileged Access			
25	CHIPTESTCTRL	0x1	RW	**CHIPTESTCTRL Privileged Access**
	CHIPTESTCTRL Privileged Access			
24	I2C1	0x1	RW	**I2C1 Privileged Access**
	I2C1 Privileged Access			
23	BURTC	0x1	RW	**BURTC Privileged Access**
	BURTC Privileged Access			
22	PIXELRZ0	0x1	RW	**PIXELRZ0 Privileged Access**
	PIXELRZ0 Privileged Access			
21	L2ICACHE0	0x1	RW	**L2ICACHE0 Privileged Access**
	L2ICACHE0 Privileged Access			
20	ETAMPDET	0x1	RW	**ETAMPDET Privileged Access**
	ETAMPDET Privileged Access			
19	TIMER3	0x1	RW	**TIMER3 Privileged Access**
	TIMER3 Privileged Access			
18	TIMER2	0x1	RW	**TIMER2 Privileged Access**
	TIMER2 Privileged Access			
17	TIMER1	0x1	RW	**TIMER1 Privileged Access**
	TIMER1 Privileged Access			

Bit	Name	Reset	Access	Description
16	TIMER0	0x1	RW	**TIMER0 Privileged Access**
	TIMER0 Privileged Access			
15	LDMAXBAR0	0x1	RW	**LDMAXBAR0 Privileged Access**
	LDMAXBAR0 Privileged Access			
14	GPIO	0x1	RW	**GPIO Privileged Access**
	GPIO Privileged Access			
13	PRS	0x1	RW	**PRS Privileged Access**
	PRS Privileged Access			
12	L1ICACHE0	0x1	RW	**L1ICACHE0 Privileged Access**
	L1ICACHE0 Privileged Access			
11	SEMAPHORE1	0x1	RW	**SEMAPHORE1 Privileged Access**
	SEMAPHORE1 Privileged Access			
10	SEMAPHORE0	0x1	RW	**SEMAPHORE0 Privileged Access**
	SEMAPHORE0 Privileged Access			
9	ULFRCO	0x1	RW	**ULFRCO Privileged Access**
	ULFRCO Privileged Access			
8	LFRCO	0x1	RW	**LFRCO Privileged Access**
	LFRCO Privileged Access			
7	LFXO	0x1	RW	**LFXO Privileged Access**
	LFXO Privileged Access			
6	DPLL0	0x1	RW	**DPLL0 Privileged Access**
	DPLL0 Privileged Access			
5	FSRCO	0x1	RW	**FSRCO Privileged Access**
	FSRCO Privileged Access			
4	LDMA0	0x1	RW	**LDMA0 Privileged Access**
	LDMA0 Privileged Access			
3	HFRCO0	0x1	RW	**HFRCO0 Privileged Access**
	HFRCO0 Privileged Access			
2	DMEM	0x1	RW	**DMEM Privileged Access**
	DMEM Privileged Access			
1	CMU	0x1	RW	**CMU Privileged Access**
	CMU Privileged Access			
0	SOCPLL0	0x1	RW	**SOCPLL0 Privileged Access**
	SOCPLL0 Privileged Access			

10.7.6 SMU_PPUNSPATD1 - PPU Privileged Access

Offset	Bit Position																															
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name	HOSTPORTAL	LPW0PORTAL	SEPORTAL	EMUCFGNS	EMU	SCRATCHPAD	SYSRTC0	AHBRADIO	QSPI0MANUAL	SEMAILBOX	SMUCFGNS	SMU	LPWAESDMACTRL	SYMCRYPTODMACTRL	EUSART0	WDOG0	I2C0	HFXO0	HFRCOEM23	PCNT0	AMUXCP0	ACMP1	ACMP0	ADC0	LETIMER0	DEVINFO	KSU	RPA	WDOG1	I2C2	PIXELRZ1	EUSART2

Bit	Name	Reset	Access	Description
31	HOSTPORTAL	0x1	RW	**HOSTPORTAL Privileged Access**
	HOSTPORTAL Privileged Access			
30	LPW0PORTAL	0x1	RW	**LPW0PORTAL Privileged Access**
	LPW0PORTAL Privileged Access			
29	SEPORTAL	0x1	RW	**SEPORTAL Privileged Access**
	SEPORTAL Privileged Access			
28	EMUCFGNS	0x1	RW	**EMUCFGNS Privileged Access**
	EMUCFGNS Privileged Access			
27	EMU	0x1	RW	**EMU Privileged Access**
	EMU Privileged Access			
26	SCRATCHPAD	0x1	RW	**SCRATCHPAD Privileged Access**
	SCRATCHPAD Privileged Access			
25	SYSRTC0	0x1	RW	**SYSRTC0 Privileged Access**
	SYSRTC0 Privileged Access			
24	AHBRADIO	0x1	RW	**AHBRADIO Privileged Access**
	AHBRADIO Privileged Access			
23	QSPI0MANUAL	0x1	RW	**QSPI0MANUAL Privileged Access**
	QSPI0MANUAL Privileged Access			
22	SEMAILBOX	0x1	RW	**SEMAILBOX Privileged Access**
	SEMAILBOX Privileged Access			
21	SMUCFGNS	0x1	RW	**SMUCFGNS Privileged Access**
	SMUCFGNS Privileged Access			
20	SMU	0x1	RW	**SMU Privileged Access**
	SMU Privileged Access			
19	LPWAESDMACTRL	0x1	RW	**LPWAESDMACTRL Privileged Access**
	LPWAESDMACTRL Privileged Access			
18	SYMCRYPTOD-MACTRL	0x1	RW	**SYMCRYPTODMACTRL Privileged Access**
	SYMCRYPTODMACTRL Privileged Access			

Bit	Name	Reset	Access	Description
17	EUSART0	0x1	RW	**EUSART0 Privileged Access**
	EUSART0 Privileged Access			
16	WDOG0	0x1	RW	**WDOG0 Privileged Access**
	WDOG0 Privileged Access			
15	I2C0	0x1	RW	**I2C0 Privileged Access**
	I2C0 Privileged Access			
14	HFXO0	0x1	RW	**HFXO0 Privileged Access**
	HFXO0 Privileged Access			
13	HFRCOEM23	0x1	RW	**HFRCOEM23 Privileged Access**
	HFRCOEM23 Privileged Access			
12	PCNT0	0x1	RW	**PCNT0 Privileged Access**
	PCNT0 Privileged Access			
11	AMUXCP0	0x1	RW	**AMUXCP0 Privileged Access**
	AMUXCP0 Privileged Access			
10	ACMP1	0x1	RW	**ACMP1 Privileged Access**
	ACMP1 Privileged Access			
9	ACMP0	0x1	RW	**ACMP0 Privileged Access**
	ACMP0 Privileged Access			
8	ADC0	0x1	RW	**ADC0 Privileged Access**
	ADC0 Privileged Access			
7	LETIMER0	0x1	RW	**LETIMER0 Privileged Access**
	LETIMER0 Privileged Access			
6	DEVINFO	0x1	RW	**DEVINFO Privileged Access**
	DEVINFO Privileged Access			
5	KSU	0x1	RW	**KSU Privileged Access**
	KSU Privileged Access			
4	RPA	0x1	RW	**RPA Privileged Access**
	RPA Privileged Access			
3	WDOG1	0x1	RW	**WDOG1 Privileged Access**
	WDOG1 Privileged Access			
2	I2C2	0x1	RW	**I2C2 Privileged Access**
	I2C2 Privileged Access			
1	PIXELRZ1	0x1	RW	**PIXELRZ1 Privileged Access**
	PIXELRZ1 Privileged Access			
0	EUSART2	0x1	RW	**EUSART2 Privileged Access**
	EUSART2 Privileged Access			

10.7.7 SMU_PPUNSFS - PPU Fault Status

Offset	Bit Position																													
0x140	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									PPUFSPERIPHID							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	PPUFSPERIPHID	0x0	R	**Peripheral I**
	ID of the peripheral that caused the fault.			

10.7.8 SMU_PPUNSFSADDR - PPU Fault Status Address

Offset	Bit Position																													
0x144	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	PPUFSADDR																															

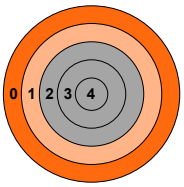
Bit	Name	Reset	Access	Description
31:0	PPUFSADDR	0x0	R	**Fault Address**
	Access address that triggered fault			

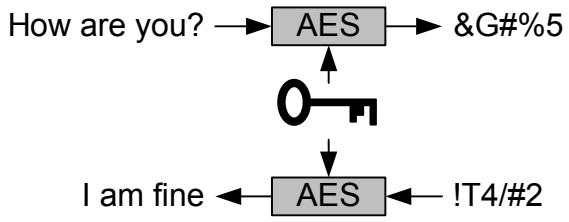
10.7.9 SMU_BMPUNSPATD0 - BMPU Privileged Attribute

Offset	Bit Position																													
0x150	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x1	0x1	0x1	0x1	0x1	0x1	0x1	0x1
Access																									RW	RW	RW	RW	RW	RW	RW	RW
Name																									SLAXICOREBUS	SYMCRYPTODMA	SEEXTDMAPORT	RPASLAXI	LDMA0PORT1	LDMA0PORT0	AHBRADIOBUS	LPWAESDMA

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	SLAXICOREBUS	0x1	RW	**SLAXICOREBUS Privileged Mode**
	SLAXICOREBUS Privileged Mode			
6	SYMCRYPTODMA	0x1	RW	**SYMCRYPTODMA Privileged Mode**
	SYMCRYPTODMA Privileged Mode			
5	SEEXTDMAPORT	0x1	RW	**SEEXTDMAPORT Privileged Mode**
	SEEXTDMAPORT Privileged Mode			
4	RPASLAXI	0x1	RW	**RPASLAXI Privileged Mode**
	RPASLAXI Privileged Mode			
3	LDMA0PORT1	0x1	RW	**LDMA0PORT1 Privileged Mode**
	LDMA0PORT1 Privileged Mode			
2	LDMA0PORT0	0x1	RW	**LDMA0PORT0 Privileged Mode**
	LDMA0PORT0 Privileged Mode			
1	AHBRADIOBUS	0x1	RW	**AHBRADIOBUS Privileged Mode**
	AHBRADIOBUS Privileged Mode			
0	LPWAESDMA	0x1	RW	**LPWAESDMA Privileged Mode**
	LPWAESDMA Privileged Mode			

11. SYMCRYPTO - Symmetric Cryptography Engine





Quick Facts

What?

The SYMCRYPTO provides fast, energy-efficient hardware for symmetric-key cryptographic operations.

Why?

Efficient hardware-based cryptography helps to meet the speed and energy demands of secure applications while providing additional protection against common attacks.

How?

Dedicated hardware allows fast processing with little to no CPU intervention.

11.1 Introduction

The SYMCRYPTO (Symmetric Cryptography Engine) includes hardware for symmetric-key cryptographic operations such as AES, a Public Key Engine (PKE) for asymmetric key operations, and a non-deterministic random number generator (RNG), suitable for key generation. SYMCRYPTO can also use the DMA for efficient data transfer into and out of the block.

11.2 Features

- Supported AES modes
  - ECB (Electronic CodeBook)
  - CBC (Cipher Block Chaining)
  - CTR (Counter)
  - CFB (Cipher FeedBack)
  - OFB (Output FeedBack)
  - CCM/CCM* (Counter mode with CBC-MAC)
  - CMAC (Cipher-based Message Authentication Code)
  - GCM/GMAC (Galois Message Authentication Mode)
- Supported hash operations
  - MD5
  - SHA-1 with hardware HMAC support
  - SHA-2 <= 512 with hardware HMAC support (e.g., digest size up to 512 bits, SHA-512)
- Supports ChaChaPoly cipher (**Note:** may not be available in certain chips)
- An internal DMA bus manager that is shared between the cryptoengines for data in/out
- Supports hardware preeemption of AES engines
- Supports key slot interface to fetch hardware keys securely

11.3 Cryptographic Functions

The SYMCRYPTO allows efficient acceleration of common cryptographic operations with a low CPU load.

Extensive software support is provided by Silicon Labs. Many security systems fail due to mistakes in the implementation. Therefore implementations should be left to experts in cryptographic algorithms. The implemented security functions are described in the documentation located at https://docs.silabs.com/.

Control and access registers for the cryptographic acceleration functions are described in the following sections. Although the register interface is provided here, it is highly recommended to use the APIs provided in the PSA Crypto library.

11.3.1 SYMCRYPTO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	SYMCRYPTO_FETCHADDR	RWH	Fetcher Address
0x008	SYMCRYPTO_FETCHLEN	RW	Fetcher Length
0x00C	SYMCRYPTO_FETCHTAG	RW	Fetcher Tag
0x010	SYMCRYPTO_PUSHADDR	RWH	Pusher Address
0x018	SYMCRYPTO_PUSHLEN	RW	Pusher Length
0x01C	SYMCRYPTO_IEN	RWH	Interrupt Enable
0x020	SYMCRYPTO_IEN_SET	W	Interrupt Enable Set
0x024	SYMCRYPTO_IEN_CLR	W	Interrupt Enable Clear
0x028	SYMCRYPTO_IF	RH INTFLAG	Interrupt Flags
0x02C	SYMCRYPTO_IFMASKED	RH	Interrupt Status
0x030	SYMCRYPTO_IF_CLR	W	Interrupt Status Clear
0x034	SYMCRYPTO_CTRL	RW	Control Register
0x038	SYMCRYPTO_CMD	W	Command Register
0x03C	SYMCRYPTO_STATUS	RH	Status Register
0x400	SYMCRYPTO_SYMCRYPTO_INCL_IPS_HW_CFG	RH	INCL_IPS_HW_CFG
0x404	SYMCRYPTO_SYMCRYPTO_BA411E_HW_CFG_1	RH	BA411E_HW_CFG_1
0x408	SYMCRYPTO_SYMCRYPTO_BA411E_HW_CFG_2	RH	BA411E_HW_CFG_2
0x40C	SYMCRYPTO_SYMCRYPTO_BA413_HW_CFG_0	RH	BA413_HW_CFG_0
0x410	SYMCRYPTO_SYMCRYPTO_BA418_HW_CFG	RH	BA418_HW_CFG
0x414	SYMCRYPTO_SYMCRYPTO_BA419_HW_CFG	RH	BA419_HW_CFG
0x418	SYMCRYPTO_SYMCRYPTO_BA424_HW_CFG	RH	BA424_HW_CFG

11.3.2 SYMCRYPTO Register Description

11.3.2.1 SYMCRYPTO_FETCHADDR - Fetcher Address

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	ADDR																															

Bit	Name	Reset	Access	Description
31:0	ADDR	0x0	RW	**Start address of data block**
	Address used by fetcher			

11.3.2.2 SYMCRYPTO_FETCHLEN - Fetcher Length

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0	0x0	0x0	0x0																											
Access		RW	RW	RW	RW																											
Name		ZPADDING	REALIGN	CONSTADDR	LENGTH																											

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30	ZPADDING	0x0	RW	**Generate zeroes**
	Generate zeroes			
29	REALIGN	0x0	RW	**Realign lengh**
	Realign length			
28	CONSTADDR	0x0	RW	**Constant address**
	Constant address			
27:0	LENGTH	0x0	RW	**Length of data block**
	Length of data block			

11.3.2.3 SYMCRYPTO_FETCHTAG - Fetcher Tag

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	TAG																															

Bit	Name	Reset	Access	Description
31:0	TAG	0x0	RW	**User tag**
	User tag			

11.3.2.4 SYMCRYPTO_PUSHADDR - Pusher Address

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	ADDR																															

Bit	Name	Reset	Access	Description
31:0	ADDR	0x0	RW	**Start address of data block**
	Start address of data block			

11.3.2.5 SYMCRYPTO_PUSHLEN - Pusher Length

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0	0x0	0x0	0x0																											
Access		RW	RW	RW	RW																											
Name		DISCARD	REALIGN	CONSTADDR	LENGTH																											

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30	DISCARD Discard data	0x0	RW	**Discard data**
29	REALIGN Realign length	0x0	RW	**Realign length**
28	CONSTADDR Constant address	0x0	RW	**Constant address**
27:0	LENGTH Start address of data block	0x0	RW	**Start address of data block**

11.3.2.6 SYMCRYPTO_IEN - Interrupt Enable

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											PUSHERERROR	PUSHERSTOPPED	PUSHERENDOFBLOCK	FETCHERERROR	FETCHERSTOPPED	FETCHERENDOFBLOCK

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	PUSHERERROR	0x0	RW	**Error interrupt enable**
	Pusher Error Interrupt Enable			
4	PUSHERSTOPPED	0x0	RW	**Stopped interrupt enable**
	Pusher Stopped Interrupt Enable			
3	PUSHERENDOF-BLOCK	0x0	RW	**End of block interrupt enable**
	Pusher End of Block Interrupt Enable			
2	FETCHERERROR	0x0	RW	**Error interrupt enable**
	Fetcher Error Interrupt Enable			
1	FETCHERSTOPPED	0x0	RW	**Stopped interrupt enable**
	Fetcher Stopped Interrupt Enable			
0	FETCHERENDOF-BLOCK	0x0	RW	**End of block interrupt enable**
	Fetcher End of Block Interrupt Enable			

11.3.2.7 SYMCRYPTO_IEN_SET - Interrupt Enable Set

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																											PUSHERERROR	PUSHERSTOPPED	PUSHERENDOFBLOCK	FETCHERERROR	FETCHERSTOPPED	FETCHERENDOFBLOCK

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	PUSHERERROR	0x0	W(nB)	**Error interrupt enable set**
	Pusher Error Interrupt Enable Set			
4	PUSHERSTOPPED	0x0	W(nB)	**Stopped interrupt enable set**
	Pusher Stopped Interrupt Enable Set			
3	PUSHERENDOF-BLOCK	0x0	W(nB)	**End of block interrupt enable set**
	Pusher End of Block Interrupt Enable Set			
2	FETCHERERROR	0x0	W(nB)	**Error interrupt enable set**
	Fetcher Error Interrupt Enable Set			
1	FETCHERSTOPPED	0x0	W(nB)	**Stopped interrupt enable set**
	Fetcher Stopped Interrupt Enable Set			
0	FETCHERENDOF-BLOCK	0x0	W(nB)	**End of block interrupt enable set**
	Fetcher End of Block Interrupt Enable Set			

11.3.2.8 SYMCRYPTO_IEN_CLR - Interrupt Enable Clear

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																											PUSHERERROR	PUSHERSTOPPED	PUSHERENDOFBLOCK	FETCHERERROR	FETCHERSTOPPED	FETCHERENDOFBLOCK

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	PUSHERERROR	0x0	W(nB)	**Error interrupt enable clear**
	Pusher Error Interrupt Enable Clear			
4	PUSHERSTOPPED	0x0	W(nB)	**Stopped interrupt enable clear**
	Pusher Stopped Interrupt Enable Clear			
3	PUSHERENDOF-BLOCK	0x0	W(nB)	**End of block interrupt enable clear**
	Pusher End of Block Interrupt Enable Clear			
2	FETCHERERROR	0x0	W(nB)	**Error interrupt enable clear**
	Fetcher Error Interrupt Enable Clear			
1	FETCHERSTOPPED	0x0	W(nB)	**Stopped interrupt enable clear**
	Fetcher Stopped Interrupt Enable Clear			
0	FETCHERENDOF-BLOCK	0x0	W(nB)	**End of block interrupt enable clear**
	Fetcher End of Block Interrupt Enable Clear			

11.3.2.9 SYMCRYPTO_IF - Interrupt Flags

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											R	R	R	R	R	R
Name																											PUSHERERROR	PUSHERSTOPPED	PUSHERENDOFBLOCK	FETCHERERROR	FETCHERSTOPPED	FETCHERENDOFBLOCK

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	PUSHERERROR	0x0	R	**Error interrupt flag**
	Triggered when an error response is received from AXI			
4	PUSHERSTOPPED	0x0	R	**Stopped interrupt flag**
	Triggered when reaching a block with Stop=1 (or end of direct transfer)			
3	PUSHERENDOF-BLOCK	0x0	R	**End of block interrupt flag**
	Triggered at the end of each block (if enabled in the descriptor - scatter-gather only)			
2	FETCHERERROR	0x0	R	**Error interrupt flag**
	Triggered when an error response is received from AXI			
1	FETCHERSTOPPED	0x0	R	**Stopped interrupt flag**
	Triggered when reaching a block with Stop=1 (or end of direct transfer)			
0	FETCHERENDOF-BLOCK	0x0	R	**End of block interrupt flag**
	Triggered at the end of each block (if enabled in the descriptor - scatter-gather only)			

11.3.2.10 SYMCRYPTO_IFMASKED - Interrupt Status

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											R	R	R	R	R	R
Name																											PUSHERERROR	PUSHERSTOPPED	PUSHERENDOFBLOCK	FETCHERERROR	FETCHERSTOPPED	FETCHERENDOFBLOCK

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	PUSHERERROR	0x0	R	**Error interrupt flag masked**
	Pusher Error Interrupt Masked by IEN			
4	PUSHERSTOPPED	0x0	R	**Stopped interrupt flag masked**
	Pusher Stopped Interrupt Masked by IEN			
3	PUSHERENDOF-BLOCK	0x0	R	**End of block interrupt flag masked**
	Pusher End of Block Interrupt Masked by IEN			
2	FETCHERERROR	0x0	R	**Error interrupt flag masked**
	Fetcher Error Interrupt Masked by IEN			
1	FETCHERSTOPPED	0x0	R	**Stopped interrupt flag masked**
	Fetcher Stopped Interrupt Masked by IEN			
0	FETCHERENDOF-BLOCK	0x0	R	**End of block interrupt flag masked**
	Fetcher End of Block Interrupt Masked by IEN			

11.3.2.11 SYMCRYPTO_IF_CLR - Interrupt Status Clear

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																											PUSHERERRORIFC	PUSHERSTOPPEDIFC	PUSHERENDOFBLOCKIFC	FETCHERERRORIFC	FETCHERSTOPPEDIFC	FETCHERENDOFBLOCKIFC

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	PUSHERERRORIFC	0x0	W(nB)	**Error interrupt flag clear**
	Pusher Error Interrupt Clear			
4	PUSHERSTOPPEDIFC	0x0	W(nB)	**Stopped interrupt flag clear**
	Pusher Stopped Interrupt Clear			
3	PUSHERENDOF-BLOCKIFC	0x0	W(nB)	**End of block interrupt flag clear**
	Pusher End of Block Interrupt Clear			
2	FETCHERERRORIFC	0x0	W(nB)	**Error interrupt flag clear**
	Fetcher Error Interrupt Clear			
1	FETCHERSTOPPE-DIFC	0x0	W(nB)	**Stopped interrupt flag clear**
	Fetcher Stopped Interrupt Clear			
0	FETCHERENDOF-BLOCKIFC	0x0	W(nB)	**End of block interrupt flag clear**
	Fetcher End of Block Interrupt Clear			

11.3.2.12 SYMCRYPTO_CTRL - Control Register

Offset	Bit Position																														
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												SWRESET	STOPPUSHER	STOPFETCHER	PUSHERSCATTERGATHER	FETCHERSCATTERGATHER

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	SWRESET	0x0	RW	**Software reset** When this bit is high, the software reset of the DMA modules, the FIFO's and the processing module will be activated. The AXI bus is not affected (pending transfers will be completed).
3	STOPPUSHER	0x0	RW	**Stop pusher** When this bit is high, the pusher will stop at the end of the current block (even if the STOP bit in the descriptor is low).
2	STOPFETCHER	0x0	RW	**Stop fetcher** When this bit is high, the fetcher will stop at the end of the current block (even if the STOP bit in the descriptor is low).
1	PUSHERSCATTER-GATHER	0x0	RW	**Pusher scatter/gather** When this bit is zero, the pusher runs in direct mode. When this bit is one, the pusher runs in scatter-gather mode.
0	FETCHERSCATTER-GATHER	0x0	RW	**Fetcher scatter/gather** When this bit is zero, the fetcher runs in direct mode. When this bit is one, the fetcher runs in scatter-gather mode.

11.3.2.13 SYMCRYPTO_CMD - Command Register

Offset	Bit Position																													
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															W	W
Name																															STARTPUSHER	STARTFETCHER

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	STARTPUSHER	0x0	W	**Start push**
	Writing a '1' starts the pusher. Writing a '0' has no effect.			
0	STARTFETCHER	0x0	W	**Start fetch**
	Writing a '1' starts the fetcher. Writing a '0' has no effect.			

11.3.2.14 SYMCRYPTO_STATUS - Status Register

Offset	Bit Position																													
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																									0x0	0x0	0x0			0x0	0x0
Access	R																									R	R	R			R	R
Name	FIFODATANUM																									SOFTRSTBSY	WAITING	NOTEMPTY			PUSHERBSY	FETCHERBSY

Bit	Name	Reset	Access	Description
31:16	FIFODATANUM	0x0	R	**Number of data in output FIFO**
	Number of bytes in output FIFO (pusher).			
15:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	SOFTRSTBSY	0x0	R	**Software reset busy**
	This bit is high when the soft reset is on-going.			
5	WAITING	0x0	R	**Pusher waiting for FIFO**
	This bit is high when the pusher if waiting for more data in output FIFO.			
4	NOTEMPTY	0x0	R	**Not empty flag from input FIFO (fetcher)**
	Not empty flag from input FIFO (fetcher).			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	PUSHERBSY	0x0	R	**Pusher busy**
	This bit is high as long as the pusher is busy.			
0	FETCHERBSY	0x0	R	**Fetcher busy**
	This bit is high as long as the fetcher is busy.			

11.3.2.15 SYMCRYPTO_SYMCRYPTO_INCL_IPS_HW_CFG - INCL_IPS_HW_CFG

Offset	Bit Position																													
0x400	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x1	0x0	0x0	0x0	0x1
Access																		R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Name																		g_IncludeAria	g_IncludeKasumi	g_IncludeSnow3G	g_IncludeHPChaChaPoly	g_IncludeNDRNG	g_IncludePKE	g_IncludeSM4	g_IncludeZUC	g_IncludeSHA3	g_IncludeChachaPoly	g_IncludeHASH	g_IncludeDES	g_IncludeAESXTS	g_IncludeAESGCM	g_IncludeAES

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14	g_IncludeAria	0x0	R	**Includes Aria**
	BA424-Aria IP included if set			
13	g_IncludeKasumi	0x0	R	**Includes Kasumi**
	BA422-Kasumi IP included if set			
12	g_IncludeSnow3G	0x0	R	**Includes Snow3G**
	BA423-Snow3G IP included if set			
11	g_IncludeHPChaChaP-oly	0x0	R	**Includes HP-ChaChaPoly**
	BA420-HP-ChaChaPoly IP included if set			
10	g_IncludeNDRNG	0x0	R	**Includes NDRNG**
	BA431 – NDRNG IP included if set			
9	g_IncludePKE	0x0	R	**Includes PKE**
	BA414EP – PKE IP included if set			
8	g_IncludeSM4	0x0	R	**Includes SM4**
	BA419 – SM4 IP included if set			
7	g_IncludeZUC	0x0	R	**Includes ZUC**
	BA421 – ZUC IP included if set			
6	g_IncludeSHA3	0x0	R	**Includes SHA3**
	BA418 – SHA3 IP included if set			
5	g_IncludeChachaPoly	0x0	R	**Includes ChachaPoly**
	BA417 – ChachaPoly IP included if set. Reset value may be different between module instances.			
4	g_IncludeHASH	0x1	R	**Includes HASH**
	BA413 – HASH IP included if set. Reset value may be different between module instances.			
3	g_IncludeDES	0x0	R	**Includes DES**
	BA412 – DES IP included if set			
2	g_IncludeAESXTS	0x0	R	**Includes AESXTS**
	BA416 – AESXTS IP included if set			

Bit	Name	Reset	Access	Description
1	g_IncludeAESGCM	0x0	R	**Includes AESGCM**
	BA415 – AESGCM IP included if set			
0	g_IncludeAES	0x1	R	**Includes AES**
	BA411E – AES IP included if set			

11.3.2.16 SYMCRYPTO_SYMCRYPTO_BA411E_HW_CFG_1 - BA411E_HW_CFG_1

Offset	Bit Position																														
0x404	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0	0x1	0x7									0x1	0x1								0x17F								
Access				R	R	R									R	R								R								
Name				g_GlitchProtection	g_CxSwitch	g_Keysize									g_UseMasking	g_CS								g_AesModesPoss								

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	g_GlitchProtection	0x0	R	**Generic g_GlitchProtection value**
	BA411E-AES engine configuration. See data sheet for details. Reset value may be different between module instances.			
27	g_CxSwitch	0x1	R	**Generic g_CxSwitch value**
	BA411E-AES engine configuration. See data sheet for details			
26:24	g_Keysize	0x7	R	**Generic g_Keysize value**
	BA411E AES engine configuration. See data sheet for details.			
23:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	g_UseMasking	0x1	R	**Generic g_UseMasking value**
	BA411E AES engine configuration. See data sheet for details. Reset value may be different between module instances.			
16	g_CS	0x1	R	**Generic g_CS value**
	BA411E AES engine configuration. See data sheet for details.			
15:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	g_AesModesPoss	0x17F	R	**AES Modes Supported**
	BA411E AES engine configuration. See data sheet for details.			

11.3.2.17 SYMCRYPTO_SYMCRYPTO_BA411E_HW_CFG_2 - BA411E_HW_CFG_2

Offset	Bit Position																													
0x408	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0				0x1								0x80															
Access					R				R								R															
Name					g_IKG_nb_AES_keys				g_Ext_nb_AES_keys								g_CtrSize															

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	g_IKG_nb_AES_keys	0x0	R	**Generic g_IKG_nb_AES_keys**
	BA411E AES engine configuration. See data sheet for details.			
23:20	g_Ext_nb_AES_keys	0x1	R	**Generic g_Ext_nb_AES_keys**
	BA411E AES engine configuration. See data sheet for details.			
19:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	g_CtrSize	0x80	R	**Generic g_CtrSize value**
	BA411E AES engine configuration. See data sheet for details.			

11.3.2.18 SYMCRYPTO_SYMCRYPTO_BA413_HW_CFG_0 - BA413_HW_CFG_0

Offset	Bit Position																													
0x40C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0				0x1					0x0	0x1	0x1										0x7F						
Access					R				R					R	R	R										R						
Name					g_IKG_nb_Hash_keys				g_Ext_nb_Hash_keys					g_HashVerifyDigest	g_HMAC_enabled	g_HashPadding										g_HashMaskFunc						

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	g_IKG_nb_Hash_keys	0x0	R	**Generic g_IKG_nb_Hash_keys value**
	BA413 Hash engine configuration. See data sheet for details.			
23:20	g_Ext_nb_Hash_keys	0x1	R	**Generic g_Ext_nb_Hash_keys value**
	BA413 Hash engine configuration. See data sheet for details.			
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	g_HashVerifyDigest	0x0	R	**Generic g_HashVerifyDigest value**
	BA413 Hash engine configuration. See data sheet for details.			
17	g_HMAC_enabled	0x1	R	**Generic g_HMAC_enabled value**
	BA413 Hash engine configuration. See data sheet for details.			
16	g_HashPadding	0x1	R	**Generic g_HashPadding value**
	BA413 Hash engine configuration. See data sheet for details.			
15:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:0	g_HashMaskFunc	0x7F	R	**Generic g_HashMaskFunc value**
	BA413 Hash engine configuration. See data sheet for details.			

11.3.2.19 SYMCRYPTO_SYMCRYPTO_BA418_HW_CFG - BA418_HW_CFG

Offset	Bit Position																															
0x410	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0				0x1					0x0	0x1																	0x1
Access					R				R					R	R																	R
Name					g_IKG_nb_Hash_keys				g_Ext_nb_Hash_keys					g_HashVerifyDigest	g_HMAC_enabled																	g_Sha3CtxtEn

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	g_IKG_nb_Hash_keys	0x0	R	**Generic g_IKG_nb_Hash_keys value** BA418 SHA3 configuration. See data sheet for details.
23:20	g_Ext_nb_Hash_keys	0x1	R	**Generic g_Ext_nb_Hash_keys value** BA418 SHA3 configuration. See data sheet for details.
19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	g_HashVerifyDigest	0x0	R	**Generic g_HashVerifyDigest value** BA418 SHA3 configuration. See data sheet for details.
17	g_HMAC_enabled	0x1	R	**Generic g_HMAC_enabled value** BA418 SHA3 configuration. See data sheet for details.
16:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	g_Sha3CtxtEn	0x1	R	**Generic g_Sha3CtxtEn value** BA418 SHA3 configuration. See data sheet for details.

11.3.2.20 SYMCRYPTO_SYMCRYPTO_BA419_HW_CFG - BA419_HW_CFG

Offset	Bit Position																															
0x414	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0									0x0								
Access															R									R								
Name															g_SM4UseMasking									g_SM4ModesPoss								

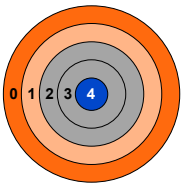
Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	g_SM4UseMasking	0x0	R	**Generic g_SM4UseMasking value**
	BA419 SM4 engine configuration. See data sheet for details.			
16:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	g_SM4ModesPoss	0x0	R	**Generic g_SM4ModesPoss value**
	BA419 SM4 engine configuration. See data sheet for details.			

11.3.2.21 SYMCRYPTO_SYMCRYPTO_BA424_HW_CFG - BA424_HW_CFG

Offset	Bit Position																															
0x418	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0								
Access																								R								
Name																								g_aria_modePoss								

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	g_aria_modePoss	0x0	R	**Generic g_aria_modePoss value**
	BA424-Aria engine configuration. See data sheet for details			

12. ETAMPDET - External Tamper Detect



Quick Facts

What?

ETAMPDET can alert the system to unauthorized physical access with very low current consumption

Why?
Detecting physical tampering can enhance security.

How?

Send low frequency random digital pulse out one pin and detect them returning on another.

12.1 Introduction

The External Tamper Detect (ETAMPDET) block functions to detect external physical tampering. For example, this feature can alert the system of an unauthorized opening of the surrounding enclosure, and the system can automatically initiate a pre-programmed response. Common implementations are to route the signal through switches that open when an enclosure is opened, or through a meander trace covering sensitive areas of PCB such that accessing the PCB requires breaking the trace thus breaking the conduction path. The ETAMPDET system uses a linear feedback shift register (LFSR) to send a time-varying pseudo random bit sequence (PRBS) digital waveform out on an ETAMPOUTx pin, and monitors a signal returned on a corresponding ETAMPINx pin. Using a random bit sequence for this makes it more difficult to thwart detection. These pins are electrically connected by an external conduction path which must be broken to gain physical access to the system. The returned PRBS signal is checked. If it is missing or does not match the sent PRBS (due to a broken trace between the IO pads), a tamper interrupt is sent and/or a PRS signal is triggered. The system designer has several choices of how to respond to alerts from ETAMPDET.

External tamper protection is useful during storage and other long periods of inactivity. This ETAMPDET feature is available in energy modes down to EM4, and current consumption is minimized to extend battery life while using this feature.

The pin pairs which can be used for this feature are limited, and are shown in GPIO Alternate Function Table of the data sheet.

12.2 Features

- Generates and compares time-varying PRBS waveform that runs down to EM4 mode
- Supports a distinct waveform per channel
- Supports software triggered reloading of LFSR
- Clock source
  - EM4GRPACLK (supports LFXO, LFRCO, ULFRCO as clock sources)
- Mismatch in waveform compare triggers
  - Wakeup event from EM0, EM1, and EM4
  - Interrupt (individual interrupt flag per channel)
  - Tamper event signal
- Supports configurable clock divider up to divide by 4096
- Selectable 1 clock delay of compare value to support traces with propagation delay > 1 clock cycle
- Filtering against spurious tamper detection

12.3 Functional Description

A basic overview of the ETAMPDET function is shown in Figure 12.1. ETAMPDET Block Diagram. Only one channel is shown in the diagram. Each channel has individual configuration and controls. The clock prescaler circuitry and output logic (IRQ, PRS) are shared between channels.

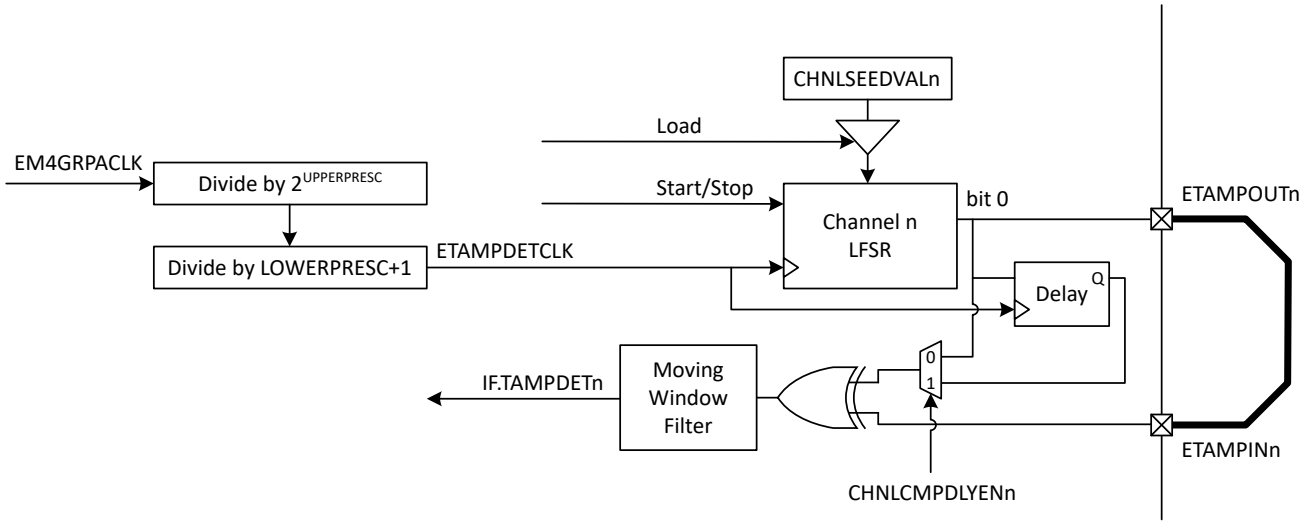


Figure 12.1. ETAMPDET Block Diagram

12.3.1 Configuration

Software will typically configure and enable the ETAMPDET block after a device reset, and leave the block running without intervention. The general procedure to enable this function is:

1. Configure the GPIO signals associated with the used channels in the GPIO block.
2. Configure the desired options and seed values for each channel used in the ETAMPDET module.
3. Write the EN.EN bit to '1' to enable the module (note that this will drive the selected ETAMPOUT pins).
4. Load the LFSR seed(s) by writing to CMD.CHNLLOADn.
5. Start the LFSR for the channel(s) by writing to CMD.CHNLSTARTn.
6. Poll the SYNCBUSY register to let the command writes take effect.

The following sections detail configuration options that are available for the ETAMPDET block. These options are generally configured before writing EN.EN, and not changed during operation.

12.3.2 Clocking

The ETAMPDET block is clocked from the EM4GRPACLK. The block has a two-stage clock prescaler which sets the desired operation frequency. The first stage is a ripple counter controlled by the UPPERPRESC field in the CLKPRESCVAL register, and the second stage is a binary counter controlled by LOWERPRESC in the CLKPRESCVAL. Figure 12.2. ETAMPDET Frequency shows the calculated frequency after the prescalers.

$$f_{ETAMPDET} = f_{EM4GRPACLK} / (2^{UPPERPRESC}) / (LOWERPRESC + 1)$$

Figure 12.2. ETAMPDET Frequency

12.3.3 Filtering

The ETAMPDET block implements a filtering mechanism to help prevent false triggers. When enabled, the filter will only trigger a tamper detect if a configurable number of mismatches has been detected within a defined number of clock cycles. This is implemented as a moving window over the collected results, counting the number of mismatches within the window and raising a tamper detected event if the allowable mismatches are exceeded.

Each channel has its own individual filter controls:

- CHNLTAMPDETFILTEN[n] in CFG register: Enables filtering for channel [n].
- CHNLFILTWINSIZE[n] in CHNLFILTWINSIZE register: Sets the moving window filter size in bits for channel [n].
- CHNLCNTMISMATCHMAX[n] in CNTMISMATCHMAX register: Sets the filter mismatch threshold in bits for channel [n].

When a tamper event has been detected, the contents of the moving window will be flushed, to avoid subsequently triggering due to the same mismatches, but will not block a new mismatch input on next clock cycle.

Figure 12.3. Moving Window Filter Example shows a sample sequence of filter inputs and the corresponding response, with a window size of 6 and a mismatch threshold of 3.

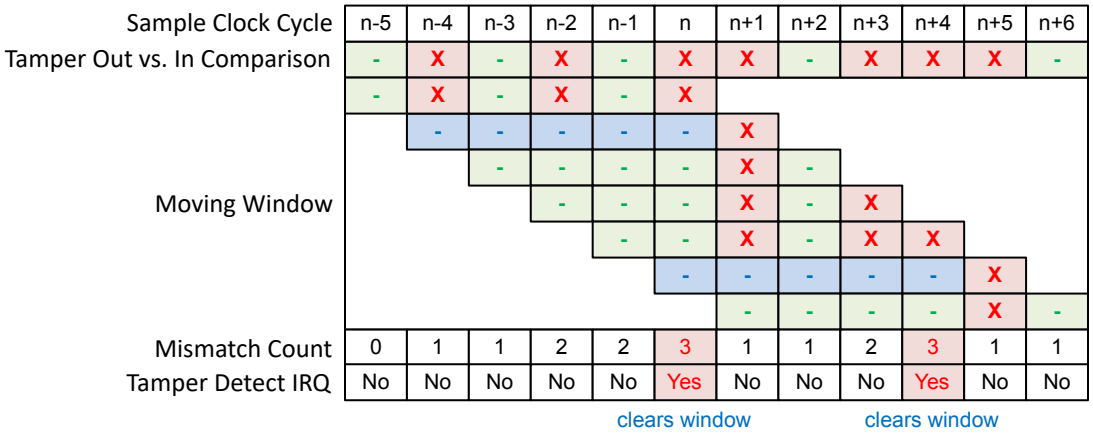


Figure 12.3. Moving Window Filter Example

12.3.4 Lock / Unlock

The ETAMPDET registers can be locked down to prevent rogue code from accidentally changing settings. Key control and configuration registers can be write-locked using the LOCK register. Reads are unaffected when the registers are locked. Writing the UNLOCK value to the LOCK register will unlock the register interface and allow access. Writing any other value to LOCK will lock the registers. The STATUS.LOCKSTATUS bit indicates whether the registers are locked.

12.3.5 Wakeup and Interrupt Sources

Each ETAMPDET channel can be individually configured to generate interrupts when a tamper event is detected, and to wake the system from low power modes down to EM4. The tamper interrupt source for each individual channel is enabled using the TAMPDETx bit in the ETAMPDET_IEN register. When a tamper event is detected, the TAMPDETx bit for the corresponding channel will be set in the IF register, and if the interrupt is enabled, an IRQ will be generated. The tamper event may also be routed via the PRS interface to other blocks in the system, or to a GPIO pin.

To enable waking from EM4, the CHNLEM4WUEN bit for the corresponding channel in the EM4WUEN register must be set to 1.

12.3.6 Commands

ETAMPDET commands can be issued through the CMD register when the module is enabled via the EN.EN bit. Three commands are available for each channel: LOAD, START, and STOP.

- A LOAD command is initiated by the CMD.CHNLLOADn bit. LOAD commands will reload the LFSR with a new seed value from the CHNLSEEDVALn register. LOAD commands can be issued before starting the counter, or while the counter is running. A LOAD command will require between 3 and 6 clocks of EM4GRPACLK and one fully divided (post-prescaler) ETAMPDET clock to take effect.
- A START command is initiated by the CMD.CHNLSTARTn bit, and will enable the LFSR shifting and tamper detection for the channel. A START command will require 6 clocks of EM4GRPACLK and one fully divided (post-prescaler) ETAMPDET clock to take effect.
- A STOP command is initiated by the CMD.CHNLSTOPn bit, and will disable the LFSR shifting and tamper detection for the channel. A STOP command will require between 3 and 6 clocks of EM4GRPACLK and one fully divided (post-prescaler) ETAMPDET clock to take effect.

The SYNCBUSY register can be polled after initiating commands to determine when the command has taken effect.

12.4 ETAMPDET Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	ETAMPDET_IPVERSION	R	IP Version ID
0x004	ETAMPDET_EN	RW ENABLE	Module Enable Register
0x00C	ETAMPDET_CFG	RW CONFIG	Configuration Register
0x010	ETAMPDET_CNTMISMATCH-MAX	RW CONFIG	Filter Threshold Register
0x014	ETAMPDET_CHNLFILTWINSIZE	RW CONFIG	Filter Moving Window Size Register
0x018	ETAMPDET_CMD	W LFSYNC	Command Register
0x01C	ETAMPDET_SYNCBUSY	RH	Syncbusy Status Register
0x020	ETAMPDET_IEN	RW	Interrupt Enable Register
0x024	ETAMPDET_IF	RWH INTFLAG	Interrupt Flag Register
0x028	ETAMPDET_STATUS	RH	Status Register
0x02C	ETAMPDET_EM4WUEN	RW CONFIG	EM4 Wakeup Request Enable Register
0x030	ETAMPDET_CHNLSEEDVAL0	RW	CHNL0 LFSR Seed Ctrl Register
0x034	ETAMPDET_CHNLSEEDVAL1	RW	CHNL1 LFSR Seed Ctrl Register
0x038	ETAMPDET_CLKPRESCVAL	RW CONFIG	Prescaler Ctrl Register
0x048	ETAMPDET_LOCK	W	Configuration Lock Register
0x1000	ETAMPDET_IPVERSION_SET	R	IP Version ID
0x1004	ETAMPDET_EN_SET	RW ENABLE	Module Enable Register
0x100C	ETAMPDET_CFG_SET	RW CONFIG	Configuration Register
0x1010	ETAMPDET_CNTMISMATCH-MAX_SET	RW CONFIG	Filter Threshold Register
0x1014	ETAMPDET_CHNLFILTWIN-SIZE_SET	RW CONFIG	Filter Moving Window Size Register
0x1018	ETAMPDET_CMD_SET	W LFSYNC	Command Register
0x101C	ETAMPDET_SYNCBUSY_SET	RH	Syncbusy Status Register
0x1020	ETAMPDET_IEN_SET	RW	Interrupt Enable Register
0x1024	ETAMPDET_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1028	ETAMPDET_STATUS_SET	RH	Status Register
0x102C	ETAMPDET_EM4WUEN_SET	RW CONFIG	EM4 Wakeup Request Enable Register
0x1030	ETAMPDET_CHNLSEEDV-AL0_SET	RW	CHNL0 LFSR Seed Ctrl Register
0x1034	ETAMPDET_CHNLSEEDV-AL1_SET	RW	CHNL1 LFSR Seed Ctrl Register
0x1038	ETAMPDET_CLKPRESCV-AL_SET	RW CONFIG	Prescaler Ctrl Register
0x1048	ETAMPDET_LOCK_SET	W	Configuration Lock Register
0x2000	ETAMPDET_IPVERSION_CLR	R	IP Version ID
0x2004	ETAMPDET_EN_CLR	RW ENABLE	Module Enable Register
0x200C	ETAMPDET_CFG_CLR	RW CONFIG	Configuration Register
0x2010	ETAMPDET_CNTMISMATCH-MAX_CLR	RW CONFIG	Filter Threshold Register

Offset	Name	Type	Description
0x2014	ETAMPDET_CHNLFILTWIN-SIZE_CLR	RW CONFIG	Filter Moving Window Size Register
0x2018	ETAMPDET_CMD_CLR	W LFSYNC	Command Register
0x201C	ETAMPDET_SYNCBUSY_CLR	RH	Syncbusy Status Register
0x2020	ETAMPDET_IEN_CLR	RW	Interrupt Enable Register
0x2024	ETAMPDET_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2028	ETAMPDET_STATUS_CLR	RH	Status Register
0x202C	ETAMPDET_EM4WUEN_CLR	RW CONFIG	EM4 Wakeup Request Enable Register
0x2030	ETAMPDET_CHNLSEEDV-AL0_CLR	RW	CHNL0 LFSR Seed Ctrl Register
0x2034	ETAMPDET_CHNLSEEDV-AL1_CLR	RW	CHNL1 LFSR Seed Ctrl Register
0x2038	ETAMPDET_CLKPRESCV-AL_CLR	RW CONFIG	Prescaler Ctrl Register
0x2048	ETAMPDET_LOCK_CLR	W	Configuration Lock Register
0x3000	ETAMPDET_IPVERSION_TGL	R	IP Version ID
0x3004	ETAMPDET_EN_TGL	RW ENABLE	Module Enable Register
0x300C	ETAMPDET_CFG_TGL	RW CONFIG	Configuration Register
0x3010	ETAMPDET_CNTMISMATCH-MAX_TGL	RW CONFIG	Filter Threshold Register
0x3014	ETAMPDET_CHNLFILTWIN-SIZE_TGL	RW CONFIG	Filter Moving Window Size Register
0x3018	ETAMPDET_CMD_TGL	W LFSYNC	Command Register
0x301C	ETAMPDET_SYNCBUSY_TGL	RH	Syncbusy Status Register
0x3020	ETAMPDET_IEN_TGL	RW	Interrupt Enable Register
0x3024	ETAMPDET_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3028	ETAMPDET_STATUS_TGL	RH	Status Register
0x302C	ETAMPDET_EM4WUEN_TGL	RW CONFIG	EM4 Wakeup Request Enable Register
0x3030	ETAMPDET_CHNLSEEDV-AL0_TGL	RW	CHNL0 LFSR Seed Ctrl Register
0x3034	ETAMPDET_CHNLSEEDV-AL1_TGL	RW	CHNL1 LFSR Seed Ctrl Register
0x3038	ETAMPDET_CLKPRESCV-AL_TGL	RW CONFIG	Prescaler Ctrl Register
0x3048	ETAMPDET_LOCK_TGL	W	Configuration Lock Register

12.5 ETAMPDET Register Description

12.5.1 ETAMPDET_IPVERSION - IP Version ID

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x2																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x2	R	**IP Version ID**

12.5.2 ETAMPDET_EN - Module Enable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	When EN is cleared, DISABLING is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFOs. The CNT and PRECNT count registers are not reset during disablement.			
0	EN	0x0	RW	**ETAMPDET Enable**
	Enable the ETAMPDET to make the peripheral clock available to the module			

12.5.3 ETAMPDET_CFG - Configuration Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											CHNLPADEN1	CHNLTAMPDETFILTEN1	CHNLCMPDLYEN1	CHNLPADEN0	CHNLTAMPDETFILTEN0	CHNLCMPDLYEN0

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CHNLPADEN1	0x0	RW	**enable driving pad**
	Enables channel 1 driving pad			
	Value	Mode		Description
	0	DISABLE		Disables channel 1 driving pad
	1	ENABLE		Enables channel 1 driving pad
4	CHNLTAMPDETFIL-TEN1	0x0	RW	**enable detect filtering**
	Enables channel 1 Tamper detect filtering feature			
	Value	Mode		Description
	0	DISABLE		Disables channel 1 Tamper detect filtering feature
	1	ENABLE		Enables channel 1 Tamper detect filtering feature
3	CHNLCMPDLYEN1	0x0	RW	**enable delay for comparison**
	Enables 1 clock delay to the TX value used for comparison. This is to support trace propagation delay more than 1 clock cycle			
	Value	Mode		Description
	0	DISABLE		Disables 1 clock delay to the TX value used for comparison.
	1	ENABLE		Enables 1 clock delay to the TX value used for comparison.
2	CHNLPADEN0	0x0	RW	**enable driving pad**
	Enables channel 0 driving pad			
	Value	Mode		Description
	0	DISABLE		Disables channel 0 driving pad
	1	ENABLE		Enables channel 0 driving pad
1	CHNLTAMPDETFIL-TEN0	0x0	RW	**enable detect filtering**
	Enables channel 0 Tamper detect filtering feature			

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	DISABLE		Disables channel 0 Tamper detect filtering feature
	1	ENABLE		Enables channel 0 Tamper detect filtering feature
0	CHNLCMPDLYEN0	0x0	RW	**enable delay for comparison**
	Enables 1 clock delay to the TX value used for comparison. This is to support trace propagation delay more than 1 clock cycle			
	Value	Mode		Description
	0	X0		Disables 1 clock delay to the TX value used for comparison.
	1	X1		Enables 1 clock delay to the TX value used for comparison.

12.5.4 ETAMPDET_CNTMISMATCHMAX - Filter Threshold Register

Offset	Bit Position																															
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0			0x0		
Access																											RW			RW		
Name																											CHNLCNTMISMATCHMAX1			CHNLCNTMISMATCHMAX0		

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:3	CHNLCNTMISMATCH-MAX1	0x0	RW	**channel 1 filter counter threshold**

Channel 1 filter threshold value where a tamper detect will be triggered

Value	Mode	Description
0	DETECTFILTERTHRES-HOLD1	Detect filter raise detect flag after seeing 1 event before reset counter expire
1	DETECTFILTERTHRES-HOLD2	Detect filter raise detect flag after seeing 2 events before reset counter expire
2	DETECTFILTERTHRES-HOLD3	Detect filter raise detect flag after seeing 3 events before reset counter expire
3	DETECTFILTERTHRES-HOLD4	Detect filter raise detect flag after seeing 4 events before reset counter expire
4	DETECTFILTERTHRES-HOLD5	Detect filter raise detect flag after seeing 5 events before reset counter expire
5	DETECTFILTERTHRES-HOLD6	Detect filter raise detect flag after seeing 6 events before reset counter expire
6	DETECTFILTERTHRES-HOLD7	Detect filter raise detect flag after seeing 7 events before reset counter expire
7	DETECTFILTERTHRES-HOLD8	Detect filter raise detect flag after seeing 8 events before reset counter expire

2:0	CHNLCNTMISMATCH-MAX0	0x0	RW	**channel 0 filter counter threshold**

Channel 0 filter threshold value where a tamper detect will be triggered

Value	Mode	Description
0	DETECTFILTERTHRES-HOLD1	Detect filter raise detect flag after seeing 1 event before reset counter expire
1	DETECTFILTERTHRES-HOLD2	Detect filter raise detect flag after seeing 2 events before reset counter expire

Bit	Name	Reset	Access	Description
	2	DETECTFILTERTHRES-HOLD3		Detect filter raise detect flag after seeing 3 events before reset counter expire
	3	DETECTFILTERTHRES-HOLD4		Detect filter raise detect flag after seeing 4 events before reset counter expire
	4	DETECTFILTERTHRES-HOLD5		Detect filter raise detect flag after seeing 5 events before reset counter expire
	5	DETECTFILTERTHRES-HOLD6		Detect filter raise detect flag after seeing 6 events before reset counter expire
	6	DETECTFILTERTHRES-HOLD7		Detect filter raise detect flag after seeing 7 events before reset counter expire
	7	DETECTFILTERTHRES-HOLD8		Detect filter raise detect flag after seeing 8 events before reset counter expire

12.5.5 ETAMPDET_CHNLFILTWINSIZE - Filter Moving Window Size Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0				0x0			
Access																									RW				RW			
Name																									CHNLFILTWINSIZE1				CHNLFILTWINSIZE0			

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:4	CHNLFILTWINSIZE1	0x0	RW	**channel 1 Filter moving window size**

Channel 1 Filter moving window size

Value	Mode	Description
0	RESERVED	Reserved
1	DETECTFILTERMOVING-WINSIZE2	Detect filter moving window size is 2
2	DETECTFILTERMOVING-WINSIZE3	Detect filter moving window size is 3
3	DETECTFILTERMOVING-WINSIZE4	Detect filter moving window size is 4
4	DETECTFILTERMOVING-WINSIZE5	Detect filter moving window size is 5
5	DETECTFILTERMOVING-WINSIZE6	Detect filter moving window size is 6
6	DETECTFILTERMOVING-WINSIZE7	Detect filter moving window size is 7
7	DETECTFILTERMOVING-WINSIZE8	Detect filter moving window size is 8
8	DETECTFILTERMOVING-WINSIZE9	Detect filter moving window size is 9
9	DETECTFILTERMOVING-WINSIZE10	Detect filter moving window size is 10
10	DETECTFILTERMOVING-WINSIZE11	Detect filter moving window size is 11
11	DETECTFILTERMOVING-WINSIZE12	Detect filter moving window size is 12
12	DETECTFILTERMOVING-WINSIZE13	Detect filter moving window size is 13
13	DETECTFILTERMOVING-WINSIZE14	Detect filter moving window size is 14
14	DETECTFILTERMOVING-WINSIZE15	Detect filter moving window size is 15

Bit	Name	Reset	Access	Description
	15	DETECTFILTERMOVING-WINSIZE16		Detect filter moving window size is 16
3:0	CHNLFILTWINSIZE0 Channel 0 Filter moving window size	0x0	RW	**channel 0 Filter moving window size**

Value	Mode	Description
0	RESERVED	Reserved
1	DETECTFILTERMOVING-WINSIZE2	Detect filter moving window size is 2
2	DETECTFILTERMOVING-WINSIZE3	Detect filter moving window size is 3
3	DETECTFILTERMOVING-WINSIZE4	Detect filter moving window size is 4
4	DETECTFILTERMOVING-WINSIZE5	Detect filter moving window size is 5
5	DETECTFILTERMOVING-WINSIZE6	Detect filter moving window size is 6
6	DETECTFILTERMOVING-WINSIZE7	Detect filter moving window size is 7
7	DETECTFILTERMOVING-WINSIZE8	Detect filter moving window size is 8
8	DETECTFILTERMOVING-WINSIZE9	Detect filter moving window size is 9
9	DETECTFILTERMOVING-WINSIZE10	Detect filter moving window size is 10
10	DETECTFILTERMOVING-WINSIZE11	Detect filter moving window size is 11
11	DETECTFILTERMOVING-WINSIZE12	Detect filter moving window size is 12
12	DETECTFILTERMOVING-WINSIZE13	Detect filter moving window size is 13
13	DETECTFILTERMOVING-WINSIZE14	Detect filter moving window size is 14
14	DETECTFILTERMOVING-WINSIZE15	Detect filter moving window size is 15
15	DETECTFILTERMOVING-WINSIZE16	Detect filter moving window size is 16

12.5.6 ETAMPDET_CMD - Command Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																											CHNLLOAD1	CHNLSTOP1	CHNLSTART1	CHNLLOAD0	CHNLSTOP0	CHNLSTART0

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CHNLLOAD1	0x0	W(nB)	**Start channel 1 tamper detection**
	Set to 1 to load seed value into channel 1 LFSR			
4	CHNLSTOP1	0x0	W(nB)	**Stop channel 1 tamper detection**
	Stop channel 1 tamper detection			
3	CHNLSTART1	0x0	W(nB)	**Start channel 1 tamper detection**
	Start channel 1 tamper detection			
2	CHNLLOAD0	0x0	W(nB)	**Start channel 0 tamper detection**
	Set to 1 to load seed value into channel 0 LFSR			
1	CHNLSTOP0	0x0	W(nB)	**Stop channel 0 tamper detection**
	Stop channel 0 tamper detection			
0	CHNLSTART0	0x0	W(nB)	**Start channel 0 tamper detection**
	Start channel 0 tamper detection			

12.5.7 ETAMPDET_SYNCBUSY - Syncbusy Status Register

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											R	R	R	R	R	R
Name																											CHNLLOAD1	CHNLSTOP1	CHNLSTART1	CHNLLOAD0	CHNLSTOP0	CHNLSTART0

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CHNLLOAD1	0x0	R	**Synchronizer busy status**
	Last writing of CHNL1LOAD is synchronizing to LF clock			
4	CHNLSTOP1	0x0	R	**Synchronizer busy status**
	Last writing of CHNL1STOP is synchronizing to LF clock			
3	CHNLSTART1	0x0	R	**Synchronizer busy status**
	Last writing of CHNL1START is synchronizing to LF clock			
2	CHNLLOAD0	0x0	R	**Synchronizer busy status**
	Last writing of CHNLLOAD is synchronizing to LF clock			
1	CHNLSTOP0	0x0	R	**Synchronizer busy status**
	Last writing of CHNL0STOP is synchronizing to LF clock			
0	CHNLSTART0	0x0	R	**Synchronizer busy status**
	Last writing of CHNL0START is synchronizing to LF clock			

12.5.8 ETAMPDET_IEN - Interrupt Enable Register

Offset	Bit Position																															
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															TAMPDET1	TAMPDET0

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	TAMPDET1	0x0	RW	**TAMPDET1 interrupt enable**
	Set to enable TAMPDET1 interrupt			
0	TAMPDET0	0x0	RW	**TAMPDET0 interrupt enable**
	Set to enable TAMPDET0 interrupt			

12.5.9 ETAMPDET_IF - Interrupt Flag Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															TAMPDET1	TAMPDET0

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	TAMPDET1	0x0	RW	**Tamper1 Detect Flag**
	Set when a tamper detect occurs on channel 1			
0	TAMPDET0	0x0	RW	**Tamper0 Detect Flag**
	Set when a tamper detect occurs on channel 0			

12.5.10 ETAMPDET_STATUS - Status Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																														0x0	0x0
Access	R																														R	R
Name	LOCKSTATUS																														CHNLRUNNING1	CHNLRUNNING0

Bit	Name	Reset	Access	Description
31	LOCKSTATUS	0x0	R	**Lock Status**
	Indicates the Indicates the current status of ETAMPDET Lock			

Value	Mode	Description
0	UNLOCKED	ETAMPDET registers are unlocked
1	LOCKED	ETAMPDET registers are locked

30:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CHNLRUNNING1	0x0	R	**Channel1 Running Status**
	Set when channel 1 is running			
0	CHNLRUNNING0	0x0	R	**Channel0 Running Status**
	Set when channel 0 is running			

12.5.11 ETAMPDET_EM4WUEN - EM4 Wakeup Request Enable Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															CHNLEM4WUEN1	CHNLEM4WUEN0

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	CHNLEM4WUEN1	0x0	RW	**Channel1 Tampdet EM4 Wakeup Enable**
	Set to enable EM4 wakeup when channel 1 tamper detect is set			
0	CHNLEM4WUEN0	0x0	RW	**Channel0 Tampdet EM4 Wakeup Enable**
	Set to enable EM4 wakeup when channel 0 tamper detect is set			

12.5.12 ETAMPDET_CHNLSEEDVAL0 - CHNL0 LFSR Seed Ctrl Register

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	CHNLSEEDVAL0																															

Bit	Name	Reset	Access	Description
31:0	CHNLSEEDVAL0	0x0	RW	**Channel 0 LFSR Seed Value**
	Seed value to load into channel 0 LFSR			

12.5.13 ETAMPDET_CHNLSEEDVAL1 - CHNL1 LFSR Seed Ctrl Register

Offset	Bit Position																															
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	CHNLSEEDVAL1																															

Bit	Name	Reset	Access	Description
31:0	CHNLSEEDVAL1	0x0	RW	**Channel 1 LFSR Seed Value**
	Seed value to load into channel 1 LFSR			

12.5.14 ETAMPDET_CLKPRESCVAL - Prescaler Ctrl Register

Offset	Bit Position																															
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																						0x0					0x0					
Access																						RW					RW					
Name																						UPPERPRESC					LOWERPRESC					

Bit	Name	Reset	Access	Description
31:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:8	UPPERPRESC	0x0	RW	**Upper part of divider ripple counter**
	Upper part of divider ripple counter			

Value	Mode	Description
0	BYPASS	Ripple counter divider is bypassed
1	DIVIDEBY2	Divide by 2
2	DIVIDEBY4	Divide by 4
3	DIVIDEBY8	Divide by 8
4	DIVIDEBY16	Divide by 16
5	DIVIDEBY32	Divide by 32
6	DIVIDEBY64	Divide by 64
7	RESERVED	Reserved, programing to this value is equal to programing 0x0

7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:0	LOWERPRESC	0x0	RW	**Lower part of divider binary counter**
	Lower part of divider binary counter			

Value	Mode	Description
0	BYPASS	Divider is bypassed
1	DIVIDEBY2	Divide by 2
2	DIVIDEBY3	Divide by 3
3	DIVIDEBY4	Divide by 4
4	DIVIDEBY5	Divide by 5
5	DIVIDEBY6	Divide by 6
6	DIVIDEBY7	Divide by 7
7	DIVIDEBY8	Divide by 8
8	DIVIDEBY9	Divide by 9
9	DIVIDEBY10	Divide by 10
10	DIVIDEBY11	Divide by 11
11	DIVIDEBY12	Divide by 12

Bit	Name	Reset	Access	Description
	12	DIVIDEBY13		Divide by 13
	13	DIVIDEBY14		Divide by 14
	14	DIVIDEBY15		Divide by 15
	15	DIVIDEBY16		Divide by 16
	16	DIVIDEBY17		Divide by 17
	17	DIVIDEBY18		Divide by 18
	18	DIVIDEBY19		Divide by 19
	19	DIVIDEBY20		Divide by 20
	20	DIVIDEBY21		Divide by 21
	21	DIVIDEBY22		Divide by 22
	22	DIVIDEBY23		Divide by 23
	23	DIVIDEBY24		Divide by 24
	24	DIVIDEBY25		Divide by 25
	25	DIVIDEBY26		Divide by 26
	26	DIVIDEBY27		Divide by 27
	27	DIVIDEBY28		Divide by 28
	28	DIVIDEBY29		Divide by 29
	29	DIVIDEBY30		Divide by 30
	30	DIVIDEBY31		Divide by 31
	31	DIVIDEBY32		Divide by 32
	32	DIVIDEBY33		Divide by 33
	33	DIVIDEBY34		Divide by 34
	34	DIVIDEBY35		Divide by 35
	35	DIVIDEBY36		Divide by 36
	36	DIVIDEBY37		Divide by 37
	37	DIVIDEBY38		Divide by 38
	38	DIVIDEBY39		Divide by 39
	39	DIVIDEBY40		Divide by 40
	40	DIVIDEBY41		Divide by 41
	41	DIVIDEBY42		Divide by 42
	42	DIVIDEBY43		Divide by 43
	43	DIVIDEBY44		Divide by 44
	44	DIVIDEBY45		Divide by 45
	45	DIVIDEBY46		Divide by 46
	46	DIVIDEBY47		Divide by 47
	47	DIVIDEBY48		Divide by 48
	48	DIVIDEBY49		Divide by 49
	49	DIVIDEBY50		Divide by 50
	50	DIVIDEBY51		Divide by 51
	51	DIVIDEBY52		Divide by 52
	52	DIVIDEBY53		Divide by 53

Bit	Name	Reset	Access	Description
	53	DIVIDEBY54		Divide by 54
	54	DIVIDEBY55		Divide by 55
	55	DIVIDEBY56		Divide by 56
	56	DIVIDEBY57		Divide by 57
	57	DIVIDEBY58		Divide by 58
	58	DIVIDEBY59		Divide by 59
	59	DIVIDEBY60		Divide by 60
	60	DIVIDEBY61		Divide by 61
	61	DIVIDEBY62		Divide by 62
	62	DIVIDEBY63		Divide by 63
	63	DIVIDEBY64		Divide by 64

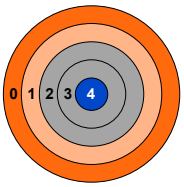
12.5.15 ETAMPDET_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0xAEE8															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0xAEE8	W	**Configuration Lock Key** Write any other value than the unlock code to lock BURTC_EN, BURTC_CFG, BURTC_CMD, BURTC_PRECNT, BURTC_CNT and BURTC_COMP registers from editing. Write the unlock code to unlock.

Value	Mode	Description
44776	UNLOCK	Write to unlock all BURTC lockable registers

13. EMU - Energy Management Unit



Quick Facts

What?

The EMU handles the different low-energy modes in SiXG301

Why?

The need for performance and peripheral functions varies over time in most applications. By efficiently scaling the available resources in real time to match the demands of the application, the energy consumption can be kept at a minimum.

How?

With a broad selection of energy modes, a high number of low-energy peripherals available, and short wake-up time, applications can dynamically minimize energy consumption during program execution.

13.1 Introduction

The Energy Management Unit (EMU) manages all the low energy modes in SiXG301. Each energy mode manages whether the CPU cores and various peripherals are available. The energy modes range from EM0 to EM4. EM0 mode provides the highest amount of features, enabling the host core subsystem and allowing all peripherals to operate. EM4 mode provides the lowest power state, with the majority of the system shut down. The EMU also controls internal power nets and voltage monitoring needed for optimal power configuration and protection.

13.2 Features

The primary features of the EMU are listed below:

- Energy Modes control
  - Entry into EM4
  - Configuration of regulators and clocks for each Energy Mode
  - Configuration of various EM4 wake-up conditions
  - Configuration of GPIO retention settings
- Power routing configurations
- Temperature sensor
- Brown Out Detection
- Reset Management
  - Power-on Reset (POR)
  - Brown-out Detection (BOD) on the following power domains:
    - Analog Unregulated Power Domain AVDD
    - Digital Unregulated Power Domain DVDD
    - I/O Unregulated Power Domain IOVDD
    - Regulated Digital Domain DECOUPLE (DEC)
  - RESETn pin reset
  - Watchdog (WDOG) reset
  - Software triggered reset (SYSRESETREQ)
  - Core LOCKUP condition
  - EM4 Detection
  - EM4 wakeup reset from GPIO pin
  - Configurable reset levels
  - A software readable register indicates the cause of the last reset

13.3 Functional Description

The EMU is responsible for managing the energy modes for the host CPU and associated peripherals that are available in SiXG301. The block works in harmony with the entire platform to easily transition between energy modes in the most efficient manner possible. A general overview of the device power distribution is shown in Figure 13.1. Power Distribution.

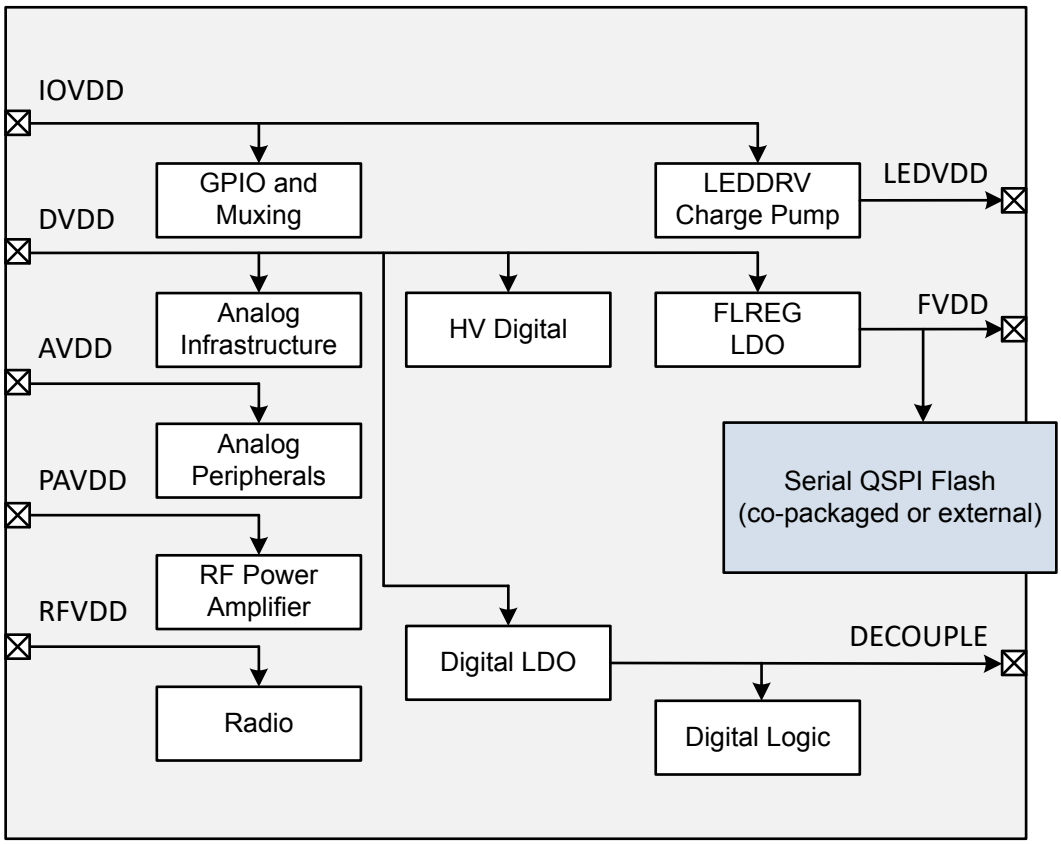


Figure 13.1. Power Distribution

The EMU is available on the peripheral bus. The energy management state machine controls the internal voltage regulators, oscillators, memories, and interrupt system. Events, interrupts, and resets can trigger the energy management state machine to return to the active state. This is further described in the following sections.

13.3.1 Energy Modes

SiXG301 features three main energy modes, referred to as Energy Mode 0 (EM0), Energy Mode 1 (EM1), and Energy Mode 4 (EM4). The Cortex-M33 is only available for program execution in EM0. In EM0 and EM1 any peripheral function can be enabled. EM4 provides a significantly reduced energy consumption while still allowing peripherals on the PDHV power domain to operate. Table 13.1. Energy Mode Transitions shows the possible transitions between different energy modes.

Table 13.1. Energy Mode Transitions

Current Mode	EM Transition Action		
	Enter EM0	Enter EM1	Enter EM4
EM0		Sleep (WFI, WFE)	EM4 Entry
EM1	IRQ		
EM4	Wake Up		

An overview of supported energy modes and available functionality is shown in the following table. For each energy mode, the system will typically default to its lowest power configuration, with non-essential clocks and peripherals disabled. Functionality may be then selectively enabled by software.

Modules with EM4 capability exist on the PDHV power domain. Refer to 13.7 Power Domains for more details.

The available energy modes are summarized in the following sections.

13.3.2 EM0

EM0 provides all system features.

- Host CPU is executing code
- Radio functionality is available
- High and low frequency clock trees are active
- All oscillators are available
- All peripheral functionality is available

13.3.3 EM1

EM1 disables the host CPU core but leaves the remaining system fully available.

- Host CPU is in sleep mode. Clocks to the core are off
- Radio functionality is available
- High and low frequency clock trees are active
- All oscillators are available
- All peripheral functionality is available

13.3.4 EM4

EM4 is the lowest energy mode of the part. There is no retention except for GPIO pin state and BURAM register values. Wake-up from EM4 requires a reset to the system, returning it back to EM0.

- Host CPU is off and powered down
- High frequency clock trees are off
- Low frequency clock trees may be active (peripheral dependent)
- No RAM is retained, except for values in BURAM
- The following oscillators are on if used by the BURTC:
  - LFRCO, LFXO, ULFRCO
- The following low frequency peripherals are available:
  - BURTC, ETAMPDET
- Wake-up to EM0 through:
  - BURTC interrupt
  - RESETn pin reset
  - Power-on reset
  - EM4WU pin interrupt
- GPIO pin states may be retained (depending on EMU_EM4CTRL.EM4IORETMODE configuration)

13.4 Entering Low Energy Modes

The following sections describe the requirements for entering the various energy modes.

13.4.1 Entry Into EM1

Energy mode EM1 is entered when the host CPU executes the Wait For Interrupt (WFI) or Wait For Event (WFE) instruction while the SLEEPDEEP bit in the Cortex-M33 System Control Register is cleared. The host CPU can re-enter sleep automatically out of an Interrupt Service Routine (ISR) if the SLEEPONEXIT bit in the Cortex-M33 System Control Register is set. Refer to ARM documentation on entering Sleep modes.

13.4.2 Entry Into EM4

Energy mode EM4 is entered through register access.

Software must ensure no modules are active when entering EM4.

Software may enter EM4 from EM0 by writing the sequence 2,3,2,3,2,3,2,3,2 to the EM4CTRL.EM4ENTRY bit field and executing a WFI instruction. If the EM4BLOCK bit in WDOGn_CFG is set, the CPU will be prevented from entering EM4 by software request.

An active debugger connection will prevent entry into EM4.

13.5 Exiting Low Energy Modes

The following sections describe how the system exits the various energy modes.

13.5.1 Exiting From EM1

Any enabled interrupt will cause the device to wake from EM1 to EM0. Upon exit from EM1, the associated ISR will be executed.

13.5.2 Exiting From EM4

Wake-up from EM4 is performed through a reset. Wake-up from a specific module must be enabled in that module's EM4WUEN register.

13.6 Power Configurations

The SiXG301 can be powered from a single supply to serve all device functions, depending on the peripherals and functions used. Note that the diagrams shown in this section provide a general picture of supply decoupling, and do not represent specific component needs. Refer to the device data sheet and evaluation card designs for component selection details.

Power Supply Pin Dependencies

Due to on-chip circuitry (e.g., diodes), some power supply pins have a dependent relationship with one or more other power supply pins. These internal relationships between the external voltages applied to the various supply pins are defined below. Exceeding the following constraints can result in damage to the device and/or increased current draw.

- DVDD ≥ DECOUPLE
- PAVDD ≥ RFVDD
- DVDD ≥ FVDD
  - DVDD must meet the minimum and maximum input supply requirements of the flash regulator when used to power FVDD.
  - In applications where FVDD is powered from an external supply, it is recommended to tie FVDD and DVDD together.
- IOVDD can be powered independently from any other supply.
  - IOVDD must meet the minimum and maximum input supply requirements of the LEDDRV charge pump when used to power LEDVDD.
- AVDD can be powered independent from any other supply.

13.6.1 Power Configuration 1: Single 3.3 V Supply

When operated from a single 3.0 - 3.6 V supply as shown in Figure 13.2. Single 3.0 - 3.6 V Supply, all device functions are available. The FLREG (flash regulator) supplies 1.8 V to a co-packaged or external serial flash device, and the LED pre-driver circuit can benefit from the LEDDRV charge pump to generate the necessary FET drive voltages.

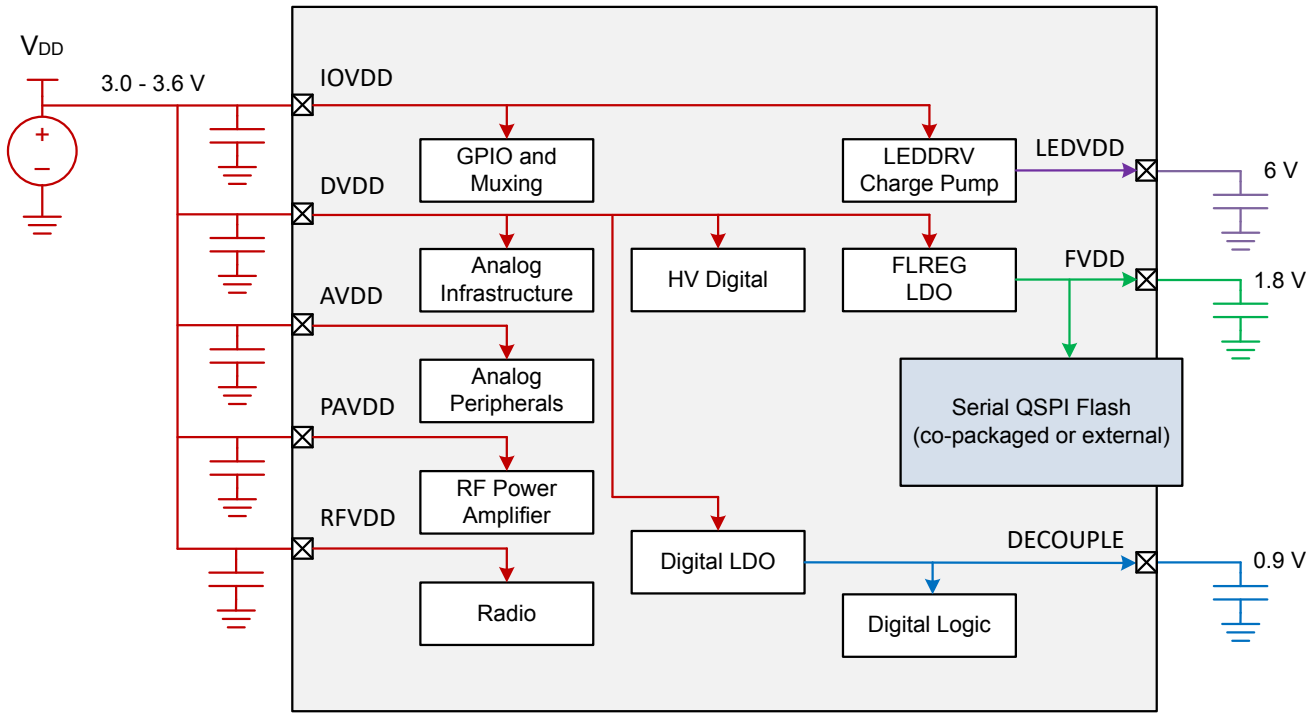


Figure 13.2. Single 3.0 - 3.6 V Supply

13.6.2 Power Configuration 2: Single 1.9 - 3.6 V Supply

When operated from a single 1.9 - 3.6 V supply as shown in Figure 13.3. Single 1.9 - 3.6 V Supply, all device functions except for the LEDDRV charge pump are available. The FLREG (flash regulator) supplies 1.8 V to a co-packaged or external serial flash device.

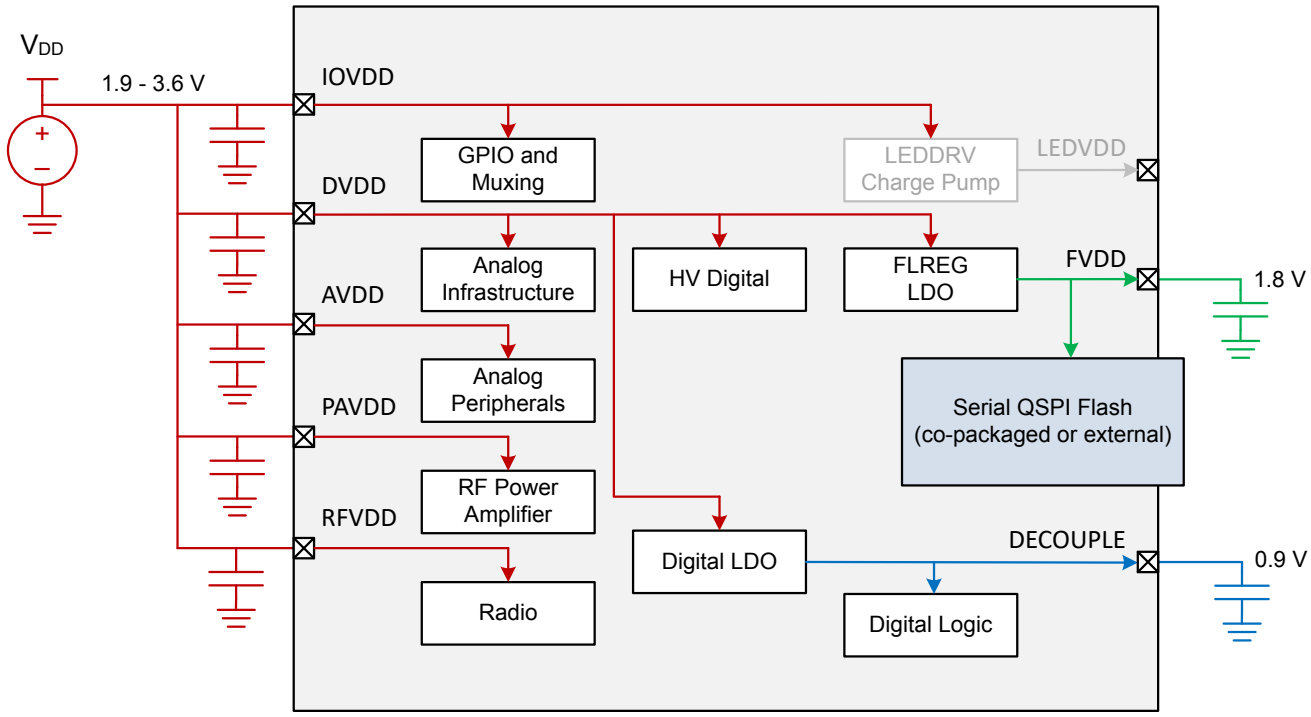


Figure 13.3. Single 1.9 - 3.6 V Supply

13.6.3 Power Configuration 3: Single 1.8 V Supply

The device can also operate from a single 1.8 V rail with ±5% tolerance, as shown in Figure 13.4. Single 1.8 V Supply. When operated at 1.8 V, the external supply should be connected to the FVDD pin. The FLREG (flash regulator) is not used in this case, and can be disabled by software on the device after boot.

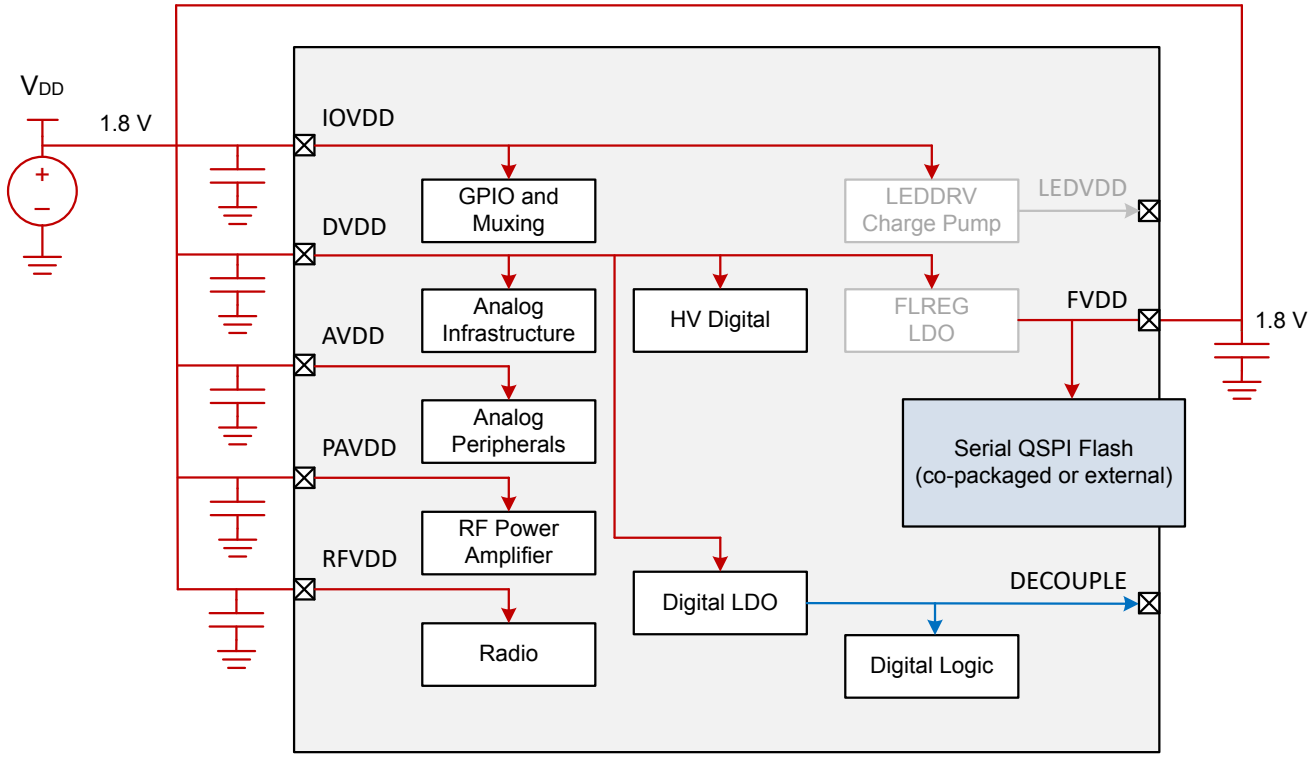


Figure 13.4. Single 1.8 V Supply

13.6.4 Other Supply Configurations

It is also possible to operate the device with multiple external supplies, so long as the supply pin dependencies in 13.6 Power Configurations are followed. Very importantly - to use the FLREG circuit, the DVDD supply must be high enough to support the FLREG LDO. When FLREG is not used, both DVDD and FVDD should be tied together and powered only from a 1.8 V supply. Refer to the device data sheet for specifics.

13.7 Power Domains

The SiXG301 device family features three primary power domains (PD0, PD1, and PDHV). Some power domains have smaller subdomains (for example, PD0A, PD0B, and so on). Power domains are managed automatically by the EMU.

The lowest-energy power domain is the "high-voltage" power domain (PDHV), which supports extremely low-energy infrastructure and peripherals. Circuits powered from PDHV are always on and available in all energy modes down to EM4.

PD1 and PD0 power most of the device circuitry, including the CPU core and all peripherals not on PDHV. PD1 and PD0 are always powered on in EM0 and EM1. PD1 and PD0 are always shut down in EM4.

Table 13.2. PDHV Peripheral Power Subdomain shows the peripherals on the PDHV domain. Any peripheral not listed is on PD1 or PD0.

Table 13.2. PDHV Peripheral Power Subdomain

Always On in EM4
PDHV
LFRCO (Non-precision mode)
LFXO
BURTC
ULFRCO
ETAMPDET
BURAM

13.8 Brown Out Detector (BOD)

Brown out detectors ensure that the minimum supply required for the chip to operate properly and safely is provided to the SiXG301. Once triggered, a BOD will generate a system reset.

All BODs detect when the supply falls below a programmed threshold except DECOVMBOD (Over Voltage Monitoring),which detects when the supply goes above a predefined threshold.

All BODs except DVDDBOD and DVDDLEBOD can be individually enabled by firmware.

Table 13.3. SiXG301 BODs

BOD	Control Register	Supported Energy Modes	Function
DVDDBOD	n/a	EM0/1	Monitors the DVDD supply in EM0 and EM1. Hardware enables this BOD automatically in EM0/EM1 and disables it in EM4.
DVDDLEBOD	n/a	EM4	Low Energy BOD monitors the DVDD supply in EM4. DVDDLEBOD is automatically masked by hardware for ~200 µs after it is enabled to allow it to settle.
DECBOD	EMU_DECBOD	EM0/1	Monitors the DECOUPLE supply. DECBOD is automatically masked by hardware for ~20 µs after it is enabled to allow it to settle.
DECOVMBOD	EMU_DECBOD	EM0/1	Monitors the DECOUPLE supply Over Voltage by detecting DECOUPLE going over a specified threshold. DECOVMBOD is automatically masked by hardware for ~20 µs after it is enabled to allow it to settle.
AVDDBOD	EMU_BOD3SENSE	EM0/1/4	Monitors the AVDD supply. Automatically masked by hardware for ~800 µs after it is enabled to allow it to settle.
IOVDDBOD	EMU_BOD3SENSE	EM0/1/4	Monitors the IOVDD supply. Automatically masked by hardware for ~800 µs after it is enabled to allow it to settle. The IOVDDBOD has two channels: IOVDD0BOD and IOVDD1BOD. IOVDD0BOD is behind an isolation barrier, and will be disabled, but indicate the supply is good under ~1.2 V. For systems where a separate supply rail is used for IOVDD and firmware must check the status, it is recommended to use the IOVDD1BOD channel.
FLBOD	EMU_FLBOD	EM0/1	Monitors the FVDD supply in EM0 and EM1. Automatically masked by hardware for ~1 µs after it is enabled to allow it to settle.

13.9 Reset Management Unit

The RMU (Reset Management Unit) inside the EMU ensures correct reset operation. It is responsible for connecting the different reset sources to the reset lines of the SiXG301. After reset, the host CPU loads the stack pointer and program entry point from memory and begins execution.

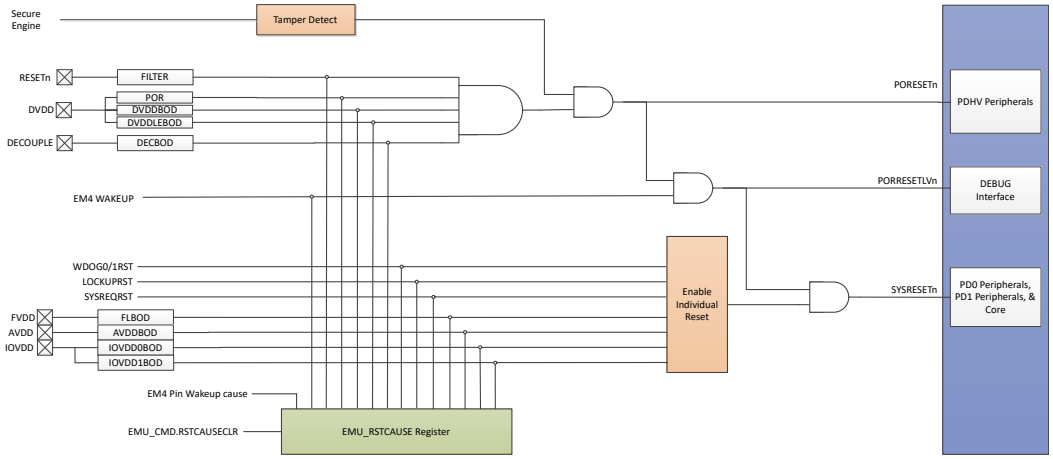


Figure 13.5. Reset Tree

There are two types of reset:

- HARD resets. Resets the entire chip. After a hard reset, the SiXG301 goes through its power up sequence and a full boot.
- SOFT resets. Resets only some of the digital low voltage logic. Resets the host CPU subsystems and peripherals but doesn't affect digital HV logic (e.g., Power control, BURTC).

SiXG301 Reset sources

- Power-on Reset (POR)
  - The POR ensures that SiXG301 does not start up before the supply voltage DVDD has reached the threshold voltage V_{DVDD_BOD} (see Device Data Sheet Electrical Characteristics for details). Before the threshold voltage is reached, SiXG301 is kept in a reset state.
- RESET pin Reset
  - The RESETn pin includes an on-chip pull-up resistor to the DVDD supply, and can therefore be left unconnected if no external reset source is needed. Also connected to the RESETn line is a filter which prevents glitches from resetting the SiXG301.
- EM4 wakeup
  - System reset following EM4 exit.
- Watchdog reset
  - The Watchdog circuit is a timer which (when enabled) must be cleared by software regularly. If software does not clear it, a Watchdog reset is activated. This functionality provides recovery from a software stalemate. Refer to the Watchdog section for specifications and description.
- Core lockup condition
  - A MCU lockup is the result of the core being locked up because of an unrecoverable exception following the activation of the processor's built-in system state protection hardware.
- Software triggered reset
  - Software may initiate a reset (e.g. if it finds itself in a non-recoverable state). By asserting the SYSRESETREQ in the Application Interrupt and Reset Control Register, a reset is issued.
- Brown-Out Detection (BOD)
  - SiXG301 has multiple built in Brown-out detection (BOD) circuits, which monitor supply voltage level during operation. BOD circuits compare supply voltage to a programmed threshold level and issue a reset request when triggered.
- Secure Engine Tamper detection
  - Secure Engine may issue a system reset request upon tamper detection.

Several of the reset sources are enabled or disabled via control bits in the EMU_RSTCTRL register.

Firmware can determine the cause of the last reset by querying the EMU_RSTCAUSE register. Once read, EMU_RSTCAUSE should be cleared via EMU_CMD.RSTCAUSECLR.

Table 13.4. Reset Sources Summary

Name	Type	Can be Disabled?	Description
POR	Hard	No	Power On Reset.
PIN	Hard	No	Pin Reset.
EM4	Soft	No	EM4 Wakeup
WDOG0	Soft	Yes	Watchdog 0
WDOG1	Soft	Yes	Watchdog 1
LOCKUP	Soft	Yes	M33 Lockup
SYSREQ	Soft	Yes	M33 Core System Reset
DVDDBOD	Hard	No	DVDD BOD
DVDDLEBOD	Hard	No	DVDD LEBOD
DECBOD	Hard	Yes	DECOUPLE BOD
AVDDBOD	Soft	Yes	AVDD BOD
IOVDD0BOD	Soft	Yes	IOVDD BOD (isolated)
IOVDD1BOD	Soft	Yes	IOVDD BOD (non-isolated)
FLBOD	Hard	Yes	Flash BOD
SETAMPER	Hard	No	SE Tamper Detection

13.10 Temperature Sensor

The EMU provides a low energy periodic temperature measurement. A temperature measurement is taken once every 250 ms, with the 9-bit result stored in TEMP bit-field in EMU_TEMP register. The temperature value is expressed in degrees Kelvin. EMU_TEMP.TEMPLSB represents the measured temperature fractional part (in ¼ degree Kelvin).

Note: The EMU temperature sensor is always periodically taking single temperature measurements, except in EM4 (shutoff) mode.

To obtain better noise resolution, the temperature sensor also implements a hardware averaging function, and averaged results can be requested using the EMU_CMD.TEMPAVGREQ command. When TEMPAVGREQ is set by software, the temperature sensor will take 16 or 64 samples as quickly as possible. The TEMPAVGNUM field in EMU_CTRL determines how many temperature measurements will be averaged. The averaged result is stored in the 11-bit field EMU_TEMP.TEMPAVG, which represents the full temperature with resolution of ¼ degree Kelvin.

The EMU provides the following features around temperature changes:

- Interrupt when temperature is updated (EMU_IF.TEMP)
- Interrupt when averaged temperature result is updated (EMU_IF.TEMPAVG)
- Interrupt from LOW level trip (generate interrupt EMU_IF.TEMPLOW when measured temperature in EMU_TEMP.TEMP is below programmed threshold EMU_TEMPLIMITS.TEMPLOW)
- Interrupt from HIGH level trip (generate interrupt EMU_IF.TEMPHIGH when measured temperature in EMU_TEMP.TEMP is above programmed threshold EMU_TEMPLIMITS.TEMPHIGH)

High and Low thresholds are specified as 9-bit degree Kelvin values and compared against the single temperature result (EMU_TEMP.TEMP).

Measured temperature can be converted to degrees Celsius by subtracting 273.15 ($T_{Celsius} = T_{Kelvin} - 273.15$).

13.10.1 Register Locks

EMU_LOCK can be used to control access to the EMU_RSTCTRL, EMU_CTRL, EMU_DECBOD, EMU_FLREG, EMU_BOD3SENSE, and EMU_FLBOD registers.

13.11 EMU Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x010	EMU_DECBOD	RW	DECOUPLE LVBOD Control Register
0x020	EMU_BOD3SENSE	RW	BOD3SENSE Control Register
0x040	EMU_FLREG	RW	FLREG Control Register
0x04C	EMU_FLBOD	RW	FLBOD Control Register
0x05C	EMU_IPVERSION	R	IP Version
0x060	EMU_LOCK	W	EMU Configuration Lock Register
0x064	EMU_IF	RWH INTFLAG	Interrupt Flags
0x068	EMU_IEN	RW	Interrupt Enables
0x06C	EMU_EM4CTRL	RW	EM4 Control
0x070	EMU_CMD	W	EMU Command Register
0x074	EMU_CTRL	RW	EMU Control Register
0x078	EMU_TEMPLIMITS	RW	EMU Temperature Thresholds
0x084	EMU_STATUS	RH	EMU Status Register
0x088	EMU_TEMP	RH	Temperature
0x090	EMU_RSTCTRL	RW	Reset Management Control Register
0x094	EMU_RSTCAUSE	RH	Reset Cause
0x1010	EMU_DECBOD_SET	RW	DECOUPLE LVBOD Control Register
0x1020	EMU_BOD3SENSE_SET	RW	BOD3SENSE Control Register
0x1040	EMU_FLREG_SET	RW	FLREG Control Register
0x104C	EMU_FLBOD_SET	RW	FLBOD Control Register
0x105C	EMU_IPVERSION_SET	R	IP Version
0x1060	EMU_LOCK_SET	W	EMU Configuration Lock Register
0x1064	EMU_IF_SET	RWH INTFLAG	Interrupt Flags
0x1068	EMU_IEN_SET	RW	Interrupt Enables
0x106C	EMU_EM4CTRL_SET	RW	EM4 Control
0x1070	EMU_CMD_SET	W	EMU Command Register
0x1074	EMU_CTRL_SET	RW	EMU Control Register
0x1078	EMU_TEMPLIMITS_SET	RW	EMU Temperature Thresholds
0x1084	EMU_STATUS_SET	RH	EMU Status Register
0x1088	EMU_TEMP_SET	RH	Temperature
0x1090	EMU_RSTCTRL_SET	RW	Reset Management Control Register
0x1094	EMU_RSTCAUSE_SET	RH	Reset Cause
0x2010	EMU_DECBOD_CLR	RW	DECOUPLE LVBOD Control Register
0x2020	EMU_BOD3SENSE_CLR	RW	BOD3SENSE Control Register
0x2040	EMU_FLREG_CLR	RW	FLREG Control Register
0x204C	EMU_FLBOD_CLR	RW	FLBOD Control Register
0x205C	EMU_IPVERSION_CLR	R	IP Version
0x2060	EMU_LOCK_CLR	W	EMU Configuration Lock Register
0x2064	EMU_IF_CLR	RWH INTFLAG	Interrupt Flags

Offset	Name	Type	Description
0x2068	EMU_IEN_CLR	RW	Interrupt Enables
0x206C	EMU_EM4CTRL_CLR	RW	EM4 Control
0x2070	EMU_CMD_CLR	W	EMU Command Register
0x2074	EMU_CTRL_CLR	RW	EMU Control Register
0x2078	EMU_TEMPLIMITS_CLR	RW	EMU Temperature Thresholds
0x2084	EMU_STATUS_CLR	RH	EMU Status Register
0x2088	EMU_TEMP_CLR	RH	Temperature
0x2090	EMU_RSTCTRL_CLR	RW	Reset Management Control Register
0x2094	EMU_RSTCAUSE_CLR	RH	Reset Cause
0x3010	EMU_DECBOD_TGL	RW	DECOUPLE LVBOD Control Register
0x3020	EMU_BOD3SENSE_TGL	RW	BOD3SENSE Control Register
0x3040	EMU_FLREG_TGL	RW	FLREG Control Register
0x304C	EMU_FLBOD_TGL	RW	FLBOD Control Register
0x305C	EMU_IPVERSION_TGL	R	IP Version
0x3060	EMU_LOCK_TGL	W	EMU Configuration Lock Register
0x3064	EMU_IF_TGL	RWH INTFLAG	Interrupt Flags
0x3068	EMU_IEN_TGL	RW	Interrupt Enables
0x306C	EMU_EM4CTRL_TGL	RW	EM4 Control
0x3070	EMU_CMD_TGL	W	EMU Command Register
0x3074	EMU_CTRL_TGL	RW	EMU Control Register
0x3078	EMU_TEMPLIMITS_TGL	RW	EMU Temperature Thresholds
0x3084	EMU_STATUS_TGL	RH	EMU Status Register
0x3088	EMU_TEMP_TGL	RH	Temperature
0x3090	EMU_RSTCTRL_TGL	RW	Reset Management Control Register
0x3094	EMU_RSTCAUSE_TGL	RH	Reset Cause

13.12 EMU Register Description

13.12.1 EMU_DECBOD - DECOUPLE LVBOD Control Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x1	0x0			0x1	0x0
Access																											RW	RW			RW	RW
Name																											DECOVMBODMASK	DECOVMBODEN			DECBODMASK	DECBODEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	DECOVMBODMASK	0x1	RW	**Over Voltage Monitor Mask**
	DECOUPLE BOD Over Voltage Monitor Mask			
4	DECOVMBODEN	0x0	RW	**Over Voltage Monitor enable**
	DECOUPLE BOD Over Voltage Monitor enable. Enables LVBOD below vref high. BOD is masked for 20us after enable			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DECBODMASK	0x1	RW	**DECBOD Mask**
	DECOUPLE BOD Mask			
0	DECBODEN	0x0	RW	**DECBOD enable**
	DECOUPLE BOD enable. Enables LVBOD above vref low. BOD is masked for 20us after enable			

13.12.2 EMU_BOD3SENSE - BOD3SENSE Control Register

Offset	Bit Position																														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														VDDIO1BODEN	VDDIO0BODEN	AVDDBODEN

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	VDDIO1BODEN	0x0	RW	**VDDIO1 BOD enable**
	BOD output is automatically masked for 100us by HW after enable is set			
1	VDDIO0BODEN	0x0	RW	**VDDIO0 BOD enable**
	BOD output is automatically masked for 100us by HW after enable is set			
0	AVDDBODEN	0x0	RW	**AVDD BOD enable**
	BOD output is automatically masked for 100us by HW after enable is set			

13.12.3 EMU_FLREG - FLREG Control Register

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0																													0x1	0x0
Access	RW	RW																													RW	RW
Name	OVROVERRIDEEN	OVRENLP																													DISCLAMP	EN

Bit	Name	Reset	Access	Description
31	OVROVERRIDEEN	0x0	RW	**Enable Override**
	Allows FLREG control override, use with OVRENLP to increase FLBOD accuracy when FLREG is disabled			
30	OVRENLP	0x0	RW	**Override Low Power Enable**
	When OVROVERRIDEEN is set, enables low power mode to increase FLBOD accuracy			
29:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISCLAMP	0x1	RW	**Disable soft pulldown**
	Soft pull down disable. Pull down should be disabled when FLREG is ON, Pull down is used for protection against overvoltage. Pull down should be disabled if Flash power is supplied by externally			
0	EN	0x0	RW	**Enable FLREG**
	Enable FLREG. Provides power to Flash and QSPI PHY. Should be disabled if Flash power is supplied by externally			

13.12.4 EMU_FLBOD - FLBOD Control Register

Offset	Bit Position																													
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x1	0x0
Access																															RW	RW
Name																															BODMASK	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	BODMASK	0x1	RW	**FLBOD Mask**
	Mask BOD output. Reset by HW in EM4			
0	EN	0x0	RW	**FLBOD Enable**
	FLBOD enable. Reset by HW in EM4			

13.12.5 EMU_IPVERSION - IP Version

Offset	Bit Position																													
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x9																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x9	R	**IP Version**
	IP Version			

13.12.6 EMU_LOCK - EMU Configuration Lock Register

Offset	Bit Position																													
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0xADE8														
Access																	W														
Name																	LOCKKEY														

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0xADE8	W	**Lock Key**
	Write any other value than the unlock code to lock			

Value	Mode	Description
44520	UNLOCK	Unlock EMU register

13.12.7 EMU_IF - Interrupt Flags

Offset	Bit Position																															
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0		0x0									0x0	0x0	0x0																
Access	RW	RW	RW		RW									RW	RW	RW																
Name	TEMPHIGH	TEMPLOW	TEMP		TEMPAVG									IOVDD1BOD	IOVDD0BOD	AVDDBOD																

Bit	Name	Reset	Access	Description
31	TEMPHIGH	0x0	RW	**Temperature high Interrupt flag**
	Measured temperature above threshold			
30	TEMPLOW	0x0	RW	**Temperature low Interrupt flag**
	Measured temperature below threshold			
29	TEMP	0x0	RW	**Temperature Interrupt flag**
	Temperature Update			
28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	TEMPAVG	0x0	RW	**Temperature Average Interrupt flag**
	Averaged Temperature Update			
26:19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	IOVDD1BOD	0x0	RW	**VDDIO1 BOD Interrupt flag**
	IOVDD1 BOD triggered			
17	IOVDD0BOD	0x0	RW	**VDDIO0 BOD Interrupt flag**
	IOVDD0 BOD triggered			
16	AVDDBOD	0x0	RW	**AVDD BOD Interrupt flag**
	AVDD BOD triggered			
15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

13.12.8 EMU_IEN - Interrupt Enables

Offset	Bit Position																															
0x068	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0		0x0									0x0	0x0	0x0																
Access	RW	RW	RW		RW									RW	RW	RW																
Name	TEMPHIGH	TEMPLOW	TEMP		TEMPAVG									IOVDD1BOD	IOVDD0BOD	AVDDBOD																

Bit	Name	Reset	Access	Description
31	TEMPHIGH	0x0	RW	**Temperature high Interrupt enable**
	Measured temperature above threshold Interrupt enable			
30	TEMPLOW	0x0	RW	**Temperature low Interrupt enable**
	Measured temperature below threshold Interrupt enable			
29	TEMP	0x0	RW	**Temperature Interrupt enable**
	Temperature Update Interrupt enable			
28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	TEMPAVG	0x0	RW	**Temperature Interrupt enable**
	Averaged Temperature Interrupt enable			
26:19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	IOVDD1BOD	0x0	RW	**VDDIO1 BOD Interrupt enable**
	IOVDD1 BOD Interrupt enable			
17	IOVDD0BOD	0x0	RW	**VDDIO0 BOD Interrupt enable**
	IOVDD0 BOD Interrupt enable			
16	AVDDBOD	0x0	RW	**AVDD BOD Interrupt enable**
	AVDD BOD Interrupt enable			
15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

13.12.9 EMU_EM4CTRL - EM4 Control

Offset	Bit Position																														
0x06C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0			0x0				0x0	
Access																								RW			RW				RW	
Name																								BOD3SENSEEM4WU			EM4IORETMODE				EM4ENTRY	

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	BOD3SENSEEM4WU	0x0	RW	**Set BOD3SENSE as EM4 wakeup**
	Enable BOD3SENSE as EM4 wakeup source			
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	EM4IORETMODE	0x0	RW	**EM4 IO retention mode**
	Determine when IO retention will be applied and removed			

Value	Mode	Description
0	DISABLE	No Retention: Pads enter reset state when entering EM4
1	EM4EXIT	Retention through EM4: Pads enter reset state when exiting EM4
2	SWUNLATCH	Retention through EM4 and Wakeup: software writes UNLATCH register to remove retention

3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	EM4ENTRY	0x0	RW	**EM4 entry request**
	This field is used to enter the Energy Mode 4 sequence. Writing the sequence 2,3,2,3,2,3,2,3,2 will enter the part into Energy Mode 4			

13.12.10 EMU_CMD - EMU Command Register

Offset	Bit Position																														
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0													0x0			0x0	
Access															W(nB)													W(nB)			W(nB)	
Name															RSTCAUSECLR													TEMPAVGREQ			EM4UNLATCH	

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	RSTCAUSECLR	0x0	W(nB)	**Reset Cause Clear**
	Set this bit to clear the RSTCAUSE register			
16:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	TEMPAVGREQ	0x0	W(nB)	**Temperature Average Request**
	Request for Averaged Temperature Measurement			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	EM4UNLATCH	0x0	W(nB)	**EM4 unlatch**
	GPIO unlatch request after EM4 wakeup. Only valid when EM4IORETMODE== SWUNLATCH			
0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

13.12.11 EMU_CTRL - EMU Control Register

Offset	Bit Position																															
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0													0x0			
Access																RW													RW			
Name																FLASHPWRUPONDEMAND													TEMPAVGNUM			

Bit	Name	Reset	Access	Description
31:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	FLASHPWRUPONDE-MAND	0x0	RW	**Enable flash on demand wakeup**
	When set, during wake up, Flash will be in power down mode until either incoming Flash data fetch or when software issue powerup command to IMEM->MSC_CMD register			
15:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	TEMPAVGNUM	0x0	RW	**Averaged Temperature samples num**
	Number of samples taken for Averaged Temperature Measurement			

Value	Mode	Description
0	N16	16 measurements
1	N64	64 measurements

2:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

13.12.12 EMU_TEMPLIMITS - EMU Temperature Thresholds

Offset	Bit Position																													
0x078	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset								0x1FF																0x0								
Access								RW																RW								
Name								TEMPHIGH																TEMPLOW								

Bit	Name	Reset	Access	Description
31:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24:16	TEMPHIGH	0x1FF	RW	**Temp High limit** Temp threshold in degree Kelvin. The TEMPHIGH interrupt flag is set when a periodic temperature measurement is equal to or higher than this value.
15:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	TEMPLOW	0x0	RW	**Temp Low limit** Temp threshold in degree Kelvin. The TEMPLOW interrupt flag is set when a periodic temperature measurement is equal to or lower than this value.

13.12.13 EMU_STATUS - EMU Status Register

Offset	Bit Position																													
0x084	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0	0x0					0x0	0x0	0x0	0x0
Access																							R	R					R	R	R	R
Name																							EM4IORET	RACACTIVE					TEMPAVGACTIVE	TEMPACTIVE	FIRSTTEMPDONE	LOCK

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	EM4IORET	0x0	R	**EM4 IO retention status**
	The status of IO retention. Will be set upon EM4 entry based on EM4IORETMODE in EMU_EM4CTRL. Cleared by setting EM4UNLATCH in EMU_CMD			
8	RACACTIVE	0x0	R	**RAC active**
	This bit indicates the status of the RAC state machine. System can not enter energy modes lower than EM1 if set.			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	TEMPAVGACTIVE	0x0	R	**Temp Average active**
	Average Temperature Measurement active			
2	TEMPACTIVE	0x0	R	**Temp active**
	Temperature Measurement active			
1	FIRSTTEMPDONE	0x0	R	**First Temp done**
	First Temperature measurement completed			
0	LOCK	0x0	R	**Lock status**
	Indicates the current status of EMU Lock			

Value	Mode	Description
0	UNLOCKED	All EMU lockable registers are unlocked.
1	LOCKED	All EMU lockable registers are locked.

13.12.14 EMU_TEMP - Temperature

Offset	Bit Position																													
0x088	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x0																0x0									0x0	
Access						R																R									R	
Name						TEMPAVG																TEMP									TEMPLSB	

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:16	TEMPAVG	0x0	R	**Averaged Temperature**
	Averaged Temperature Measurement. Temperature in Kelvin. 9 integer bits and 2 decimal bits (0.25 Degree resolution)			
15:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:2	TEMP	0x0	R	**Temperature measured**
	Temperature in Kelvin. Value of last periodic temperature measurement. Value is asynchronously updated.			
1:0	TEMPLSB	0x0	R	**Temperature measured decimal part**
	Temperature decimal part			

13.12.15 EMU_RSTCTRL - Reset Management Control Register

Offset	Bit Position																													
0x090	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x1		0x0	0x0	0x0			0x0	0x1	0x1	0x1
Access																					RW	RW		RW	RW	RW			RW	RW	RW	RW
Name																					FLBODRMODE	DECBODRMODE		IOVDD1BODRMODE	IOVDD0BODRMODE	AVDDBODRMODE			LOCKUPRMODE	SYSRMODE	WDOG1RMODE	WDOG0RMODE

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	FLBODRMODE	0x0	RW	**Enable FLBOD reset**

FLBOD Reset Mode. FLASH supply monitoring. BOD must be trimmed before it is used as a reset source.

Value	Mode	Description
0	DISABLED	Reset request is blocked
1	ENABLED	The entire device is reset

10	DECBODRMODE	0x1	RW	**Enable DECBOD reset**

LVBOD Reset Mode. DECOUPLE monitoring. BOD must be trimmed before it is used as a reset source.

Value	Mode	Description
0	DISABLED	Reset request is blocked
1	ENABLED	The entire device is reset

9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	IOVDD1BODRMODE	0x0	RW	**Enable VDDIO1 BOD reset**

LEBOD3 Reset Mode. IOVDD1 monitoring. BOD must be trimmed before it is used as a reset source.

Value	Mode	Description
0	DISABLED	Reset request is blocked
1	ENABLED	The entire device is reset except some EMU registers

7	IOVDD0BODRMODE	0x0	RW	**Enable VDDIO0 BOD reset**

LEBOD2 Reset Mode. IOVDD0 monitoring. BOD must be trimmed before it is used as a reset source.

Value	Mode	Description
0	DISABLED	Reset request is blocked
1	ENABLED	The entire device is reset except some EMU registers

6	AVDDBODRMODE	0x0	RW	**Enable AVDD BOD reset**

LEBOD1 Reset Mode. AVDD monitoring. BOD must be trimmed before it is used as a reset source.

Value	Mode	Description

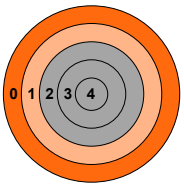
Bit	Name	Reset	Access	Description
	0	DISABLED		Reset Request is block
	1	ENABLED		The entire device is reset except some EMU registers
5:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	LOCKUPRMODE	0x0	RW	**Enable M33 Lockup reset**
	Core LOCKUP Reset Mode			
	Value	Mode		Description
	0	DISABLED		Reset Request is Block
	1	ENABLED		The entire device is reset except some EMU registers
2	SYSRMODE	0x1	RW	**Enable M33 System reset**
	Core Sysreset Reset Mode			
	Value	Mode		Description
	0	DISABLED		Reset request is blocked
	1	ENABLED		Device is reset except some EMU registers
1	WDOG1RMODE	0x1	RW	**Enable WDOG1 reset**
	WDOG1 Reset Mode			
	Value	Mode		Description
	0	DISABLED		Reset request is blocked
	1	ENABLED		The entire device is reset except some EMU registers
0	WDOG0RMODE	0x1	RW	**Enable WDOG0 reset**
	WDOG0 Reset Mode			
	Value	Mode		Description
	0	DISABLED		Reset request is blocked
	1	ENABLED		The entire device is reset except some EMU registers

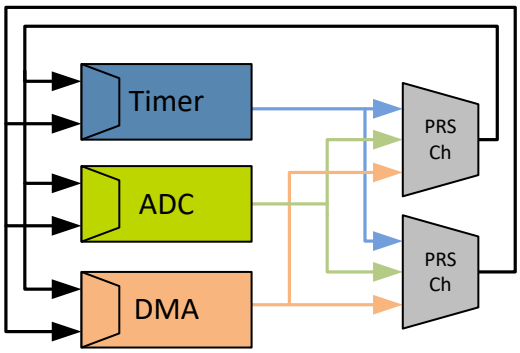
13.12.16 EMU_RSTCAUSE - Reset Cause

Offset	Bit Position																													
0x094	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																		R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Name																		SETAMPER	FLBOD	IOVDD1BOD	IOVDD0BOD	AVDDBOD	DECBOD	DVDDLEBOD	DVDDBOD	SYSREQ	LOCKUP	WDOG1	WDOG0	EM4	PIN	POR

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14	SETAMPER	0x0	R	**SE Tamper event Reset**
	Last reset was a SE Tamper event reset			
13	FLBOD	0x0	R	**FLBOD Reset**
	Brown Out Detector monitoring Flash supply			
12	IOVDD1BOD	0x0	R	**LEBOD3 Reset**
	Brown Out Detector monitoring IOVDD (non-isolated)			
11	IOVDD0BOD	0x0	R	**LEBOD2 Reset**
	Brown Out Detector monitoring IOVDD (isolated)			
10	AVDDBOD	0x0	R	**LEBOD1 Reset**
	Brown Out Detector monitoring AVDD			
9	DECBOD	0x0	R	**LVBOD Reset**
	Brown Out Detector monitoring DECOUPLE			
8	DVDDLEBOD	0x0	R	**LEBOD Reset**
	Brown Out Detector monitoring DVDD in EM4			
7	DVDDBOD	0x0	R	**HVBOD Reset**
	Brown Out Detector monitoring DVDD in EM0 and EM1			
6	SYSREQ	0x0	R	**M33 Core Sys Reset**
	Last Reset was as M33 Core System reset			
5	LOCKUP	0x0	R	**M33 Core Lockup Reset**
	Last Reset was as M33 Core Lockup reset			
4	WDOG1	0x0	R	**Watchdog 1 Reset**
	Last reset was a Watchdog 1 reset			
3	WDOG0	0x0	R	**Watchdog 0 Reset**
	Last reset was a Watchdog 0 reset			
2	EM4	0x0	R	**EM4 Wakeup Reset**
	Last reset was a EM4 Wakeup			
1	PIN	0x0	R	**Pin Reset**
	Last reset was a Pin reset			
0	POR	0x0	R	**Power On Reset**
	Last reset was a Power On Reset			

14. PRS - Peripheral Reflex System





Quick Facts

What?

The PRS allows configurable, fast, and autonomous communication between peripherals.

Why?

Events and signals from one peripheral can be used as input signals to trigger actions in other peripherals. PRS reduces latency and ensures predictable timing by reducing software overhead and thus current consumption.

How?

Without CPU intervention, the peripherals can send reflex signals to each other in single or chained steps. The peripherals can be set up to perform actions based on the incoming reflex signals. This results in improved system performance and reduced energy consumption.

14.1 Introduction

The Peripheral Reflex System is a signal routing network allowing direct communication between different peripheral modules without involving the CPU. Peripheral modules which send out reflex signals to the PRS are called producers, and modules accepting reflex signals are called consumers. The PRS routes the reflex signals from producer to consumer peripherals, which perform actions depending on the reflex signals received.

14.2 Features

12 configurable asynchronous channels

- Each channel can be connected to any producer
- Consumers can be configured to listen to any asynchronous channel
- Can generate events to the CPU and the DMA
- Software controlled channel output using the ASYNC_SWPULSE and ASYNC_SWLEVEL registers
- Configurable logic to implement combinational functions between channels; multiple channels may be cascaded to produce more complex functions

4 configurable synchronous channels

- Special set of channels for high speed signalling between TIMER and ADC blocks

14.3 Functional Description

The PRS contains 12 asynchronous and 4 synchronous reflex channels. An overview of an asynchronous PRS reflex channel is shown in Figure 14.1. PRS Asynchronous Channel Overview. Synchronous channels are similar but do not include the configurable logic block or software level / pulse features. Asynchronous channels can be connected to any signal offered by the producers while the synchronous channels are restricted to special signals from the TIMER and ADC modules.

Similarly on the consumer side, all the peripherals can listen to asynchronous channels while only the TIMER and ADC modules can listen to synchronous channels. The consumers of a channel (synchronous or asynchronous) can choose which PRS channel to listen to and perform actions based on the reflex signals routed through that channel. Synchronous channels are only available in EM0 and EM1 while asynchronous channels are available in all energy modes except EM4.

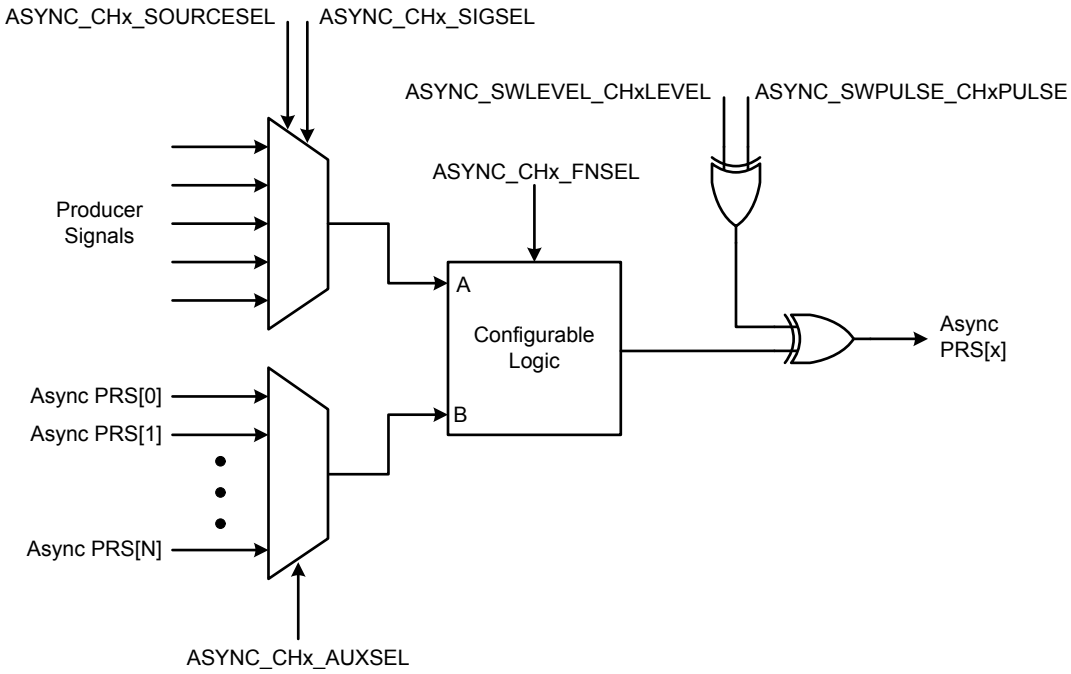


Figure 14.1. PRS Asynchronous Channel Overview

14.3.1 Asynchronous Channel Functions

Different functions can be applied to a reflex signal within the PRS. The asynchronous PRS channels can be manually triggered by writing to PRS_ASYNC_SWPULSE or PRS_ASYNC_SWLEVEL. SWLEVEL[n] is a programmable level for each asynchronous channel and holds the value it is programmed to. Setting SWPULSE[n] will cause the asynchronous channel to output a high pulse that is one PCLK clock cycle wide. The SWLEVEL[n] and SWPULSE[n] signals are then XOR'ed with the output from the configurable logic block to form the output signal and is sent to the channel selection logic for every consumer signal. For example, when SWLEVEL[n] is set, if configurable logic produces a signal of 1, this will cause a channel output of 0.

14.3.2 Configurable Logic

The configurable logic feature enables a PRS channel to perform logic operations on the signal coming from the selected producer. Every asynchronous channel has a configurable logic block that can be programmed using the FNSEL field in the asynchronous channel control register. The configurable logic block for each channel has two inputs. Input A is the signal from the selected producer determined by SOURCESEL and SIGSEL of PRS_ASYNC_CHx_CTRL. Input B may be selected from the output of any other asynchronous PRS channel using the ASYNC_CHx_CTRL.AUXSEL field. This allows for more complex logic functions to be created using multiple PRS channels.

Table 14.1. Configurable Logic Look up Table

A	B	FNSEL
0	0	FNSEL[0]
0	1	FNSEL[1]
1	0	FNSEL[2]
1	1	FNSEL[3]

The configurable logic feature is implemented as a 2 input look up table, with each bit of FNSEL representing the outcome for a specific input combination (see Table 14.1. Configurable Logic Look up Table). For example, if input A is 0 and input B is 1, then the PRS output will assume the value of bit 1 of FNSEL (FNSEL[1]).

To calculate the FNSEL field for an "A NAND B" function, the truth table can be filled out as:

Table 14.2. A NAND B Example

A	B	FNSEL = (A NAND B)
0	0	FNSEL[0] = 1
0	1	FNSEL[1] = 1
1	0	FNSEL[2] = 1
1	1	FNSEL[3] = 0

In this example, the value of FNSEL has been calculated to be 0111 (binary), or 0x7.

Using the FNSEL field, a total of 16 two-input logic functions can be implemented, as shown in Table 14.3. List of Logic Functions.

Table 14.3. List of Logic Functions

FNSEL value	Implemented Function
0x0	0
0x1	A NOR B
0x2	(NOT A) AND B
0x3	NOT A
0x4	A AND (NOT B)
0x5	(NOT B)
0x6	A XOR B
0x7	A NAND B
0x8	A AND B
0x9	A XNOR B
0xA	B
0xB	(NOT A) OR B

FNSEL value	Implemented Function
0xC	A
0xD	A OR (NOT B)
0xE	A OR B
0xF	1

The default value of FNSEL is 0xC, meaning that the input from the selected producer goes through unchanged. This feature can be used to combine multiple channels to get even more complex functions.

14.3.3 Producers

Through SOURCESEL in PRS_SYNC_CHx_CTRL or PRS_ASYNC_CHx_CTRL, each PRS channel (synchronous and asynchronous respectively) selects its signal producers. Each producer outputs one or more signals which can be selected by setting the SIGSEL field. Setting the SOURCESEL bits to 0 (Off) leads to a constant 0 output from the input mux regardless of SIGSEL.

The GPIO producer signals depend on settings in the GPIO module. They are selected using the edge interrupt configuration settings described in 25.3.10.1 Standard Interrupt Generation. PIN0 uses settings for the EXTI0 interrupt, PIN1 uses settings for EXTI1, and so on.

For example, to route PB00 as a producer for PRS channel 2, EXTI0, EXTI1, EXTI2, or EXTI3 should be configured to connect to PB00, and the corresponding GPIO PINx should be selected as the PRS channel 2 producer. If we choose EXTI1 via PRS producer "GPIO PIN1":

1. GPIO_EXTIPSELL.EXTIPSEL1 = PORTB, and GPIO_EXTIPINSELL.EXTIPINSEL1 = PIN0 connect PB00 through the EXTI1 signal.
2. PRS_ASYNC_CH2_CTRL.SOURCESEL = GPIO, and PRS_ASYNC_CH2_CTRL.SIGSEL = PIN1 connects the PIN1 (EXTI1) signal to asynchronous PRS channel 2 as a producer.

14.3.3.1 Producer Details

Table 14.4. Synchronous PRS Producers

Peripheral	SOURCESEL	Signal	SIGSEL
TIMER0	TIMER0 (0x01)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
TIMER1	TIMER1 (0x02)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
TIMER2	TIMER2 (0x03)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
		CC3	0x5
		CC4	0x6
		CC5	0x7
		CC6	0x8
TIMER3	TIMER3 (0x04)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
		CC3	0x5
		CC4	0x6
		CC5	0x7
		CC6	0x8
ADC0	ADC0 (0x05)	SCANENTRYDONE	0x0
		SCANTABLEDONE	0x1

Table 14.5. Asynchronous PRS Producers

Peripheral	SOURCESEL	Signal	SIGSEL
LETIMER0	LETIMER0 (0x01)	CH0	0x0
		CH1	0x1
BURTC	BURTC (0x02)	COMP	0x0
		OVERFLOW	0x1

Peripheral	SOURCESEL	Signal	SIGSEL
GPIO	GPIO (0x03)	PIN0	0x0
		PIN1	0x1
		PIN2	0x2
		PIN3	0x3
		PIN4	0x4
		PIN5	0x5
		PIN6	0x6
		PIN7	0x7
CMU	CMUL (0x04)	CLKOUT0	0x0
		CLKOUT1	0x1
		CLKOUT2	0x2
PRS	PRSL (0x07)	ASYNCH0	0x0
		ASYNCH1	0x1
		ASYNCH2	0x2
		ASYNCH3	0x3
		ASYNCH4	0x4
		ASYNCH5	0x5
		ASYNCH6	0x6
		ASYNCH7	0x7
	PRS (0x08)	ASYNCH8	0x0
		ASYNCH9	0x1
		ASYNCH10	0x2
		ASYNCH11	0x3
ACMP0	ACMP0 (0x09)	OUT	0x0
ACMP1	ACMP1 (0x0A)	OUT	0x0
PCNT0	PCNT0 (0x0B)	DIR	0x0
		UFOF	0x1
SYSRTC0	SYSRTC0 (0x0C)	GRP0OUT0	0x0
		GRP0OUT1	0x1
		GRP1OUT0	0x2
		GRP1OUT1	0x3
HFXO0	HFXO0L (0x0D)	STATUS	0x0
		STATUS1	0x1
EUSART0	EUSART0L (0x0F)	CS	0x0
		IRDATX	0x1
		RTS	0x2
		RXDATAV	0x3
		TX	0x4
		TXC	0x5
		RXFL	0x6
		TXFL	0x7

Peripheral	SOURCESEL	Signal	SIGSEL
ADC0	ADC0 (0x11)	SCANENTRYDONE	0x0
		SCANTABLEDONE	0x1
		DEBUG	0x2
		DEBUG1	0x3
		DEBUG2	0x4
		DEBUG3	0x5
		DEBUG4	0x6
ETAMPDET	ETAMPDET (0x12)	TAMPERSRCETAMPDET	0x0
FSRCO	FSRCO (0x13)	CLKENHV	0x0
		CLKENLV	0x1
		CLKRAWHV	0x2
		CLKRAWLV	0x3
LFRCO	LFRCO (0x16)	CALMEAS	0x0
		SDM	0x1
		TCMEAS	0x2
HFRCOEM23	HFRCOEM23 (0x17)	CLKPD1RAW	0x4
TIMER0	TIMER0 (0x20)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
TIMER1	TIMER1 (0x21)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
TIMER2	TIMER2L (0x22)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
		CC3	0x5
		CC4	0x6
		CC5	0x7
	TIMER2 (0x23)	CC6	0x0

Peripheral	SOURCESEL	Signal	SIGSEL
TIMER3	TIMER3L (0x24)	UF	0x0
		OF	0x1
		CC0	0x2
		CC1	0x3
		CC2	0x4
		CC3	0x5
		CC4	0x6
		CC5	0x7
	TIMER3 (0x25)	CC6	0x0
CORE	CORE (0x26)	CTIOUT0	0x0
		CTIOUT1	0x1
		CTIOUT2	0x2
		CTIOUT3	0x3
AGC	AGCL (0x27)	CCA	0x0
		CCAREQ	0x1
		GAINADJUST	0x2
		GAINOK	0x3
		GAINREDUCED	0x4
		IFPKI1	0x5
		IFPKQ2	0x6
		IFPKRST	0x7
	AGC (0x28)	PEAKDET	0x0
		PROPAGATED	0x1
		RSSIDONE	0x2
BUFC	BUFC (0x29)	THR0	0x0
		THR1	0x1
		THR2	0x2
		THR3	0x3

Peripheral	SOURCESEL	Signal	SIGSEL
MODEM	MODEML (0x2A)	ADVANCE	0x0
		ANT0	0x1
		ANT1	0x2
		COHDSADET	0x3
		COHDSALIVE	0x4
		DCLK	0x5
		DOUT	0x6
		FRAMEDET	0x7
	MODEM (0x2B)	FRAMESENT	0x0
		LOWCORR	0x1
		LRDSADET	0x2
		LRDSALIVE	0x3
		NEWSYMBOL	0x4
		NEWWND	0x5
		POSTPONE	0x6
		PREDET	0x7
	MODEMH (0x2C)	PRESENT	0x0
		RSSIJUMP	0x1
		SYNCSENT	0x2
		TIMDET	0x3
		WEAK	0x4
		EOF	0x5
		SI	0x6
FRC	FRC (0x2D)	DCLK	0x0
		DOUT	0x1
PROTIMER	PROTIMERL (0x2E)	BOF	0x0
		CC0	0x1
		CC1	0x2
		CC2	0x3
		CC3	0x4
		CC4	0x5
		LBTF	0x6
		LBTR	0x7
	PROTIMER (0x2F)	LBTS	0x0
		POF	0x1
		T0MATCH	0x2
		T0UF	0x3
		T1MATCH	0x4
		T1UF	0x5
		WOF	0x6

Peripheral	SOURCESEL	Signal	SIGSEL
SYNTH	SYNTH (0x30)	MUX0	0x0
		MUX1	0x1
RAC	RACL (0x31)	ACTIVE	0x0
		LNAEN	0x1
		PAEN	0x2
		RX	0x3
		TX	0x4
		CTIOUT0	0x5
		CTIOUT1	0x6
		CTIOUT2	0x7
	RAC (0x32)	CTIOUT3	0x0
		AUXADCDATA	0x1
		AUXADCDATAVALID	0x2
EUSART1	EUSART1L (0x33)	CS	0x0
		IRDATX	0x1
		RTS	0x2
		RXDATAV	0x3
		TX	0x4
		TXC	0x5
		RXFL	0x6
		TXFL	0x7
EUSART2	EUSART2L (0x35)	IRDATX	0x0
		RTS	0x1
		RXDATAV	0x2
		RXFL	0x3
		TX	0x4
		TXC	0x5
		TXFL	0x6
		CS	0x7
RFTIMER	RFTIMER (0x37)	MATCH0	0x0
		MATCH1	0x1
		OF	0x2
SEQACC	SEQACC (0x38)	TRIGGER0	0x0
		TRIGGER1	0x1
		TRIGGER2	0x2
		TRIGGER3	0x3
HFRCO0	HFRCO0 (0x39)	CLKPD1RAW	0x4
HFRCOLPW	HFRCOLPW (0x3A)	CLKPD1RAW	0x0

Peripheral	SOURCESEL	Signal	SIGSEL
SOCPLL0	SOCPLL0 (0x3E)	SOCPLLDBG0	0x0
		SOCPLLDBG1	0x1
		SOCPLLDBG2	0x2
		SOCPLLDBG3	0x3
PIXELRZ0	PIXELRZ0 (0x3F)	PIXELDBGSTATE0	0x0
		PIXELDBGSTATE1	0x1
		PIXELDBGSTATE2	0x2
		PIXELDBGSTATE3	0x3
PIXELRZ1	PIXELRZ1 (0x40)	PIXELDBGSTATE0	0x0
		PIXELDBGSTATE1	0x1
		PIXELDBGSTATE2	0x2
		PIXELDBGSTATE3	0x3
LEDDRV0	LEDDRV0 (0x41)	LEDDRVDBG0	0x0
		LEDDRVDBG1	0x1
		LEDDRVDBG2	0x2
		LEDDRVDBG3	0x3
RPA	RPA (0x42)	RPAAESDONE	0x0
		RPAAXIDONE	0x1
		RPACOREDONE	0x2
		RPAKSUDONE	0x3
KSU	KSU (0x43)	FFPROCSTART0	0x0
		FFPROCSTART1	0x1
		FFPROCSTART2	0x2
		KSUSTATE0	0x3
		KSUSTATE1	0x4
		KSUSTATE2	0x5
		KSUDONEACK	0x6
		SLAXI2RAMCTRDONEACK	0x7

14.3.4 Consumers

Consumer peripherals can be set to listen to a PRS channel and perform an action based on the signal received on that channel. This is done by programming the PRSSEL or SPRSSEL fields in the PRS_CONSUMER registers. SPRSSEL is only present for signals with the ability to listen to synchronous channels. The consumer registers follow the naming convention PRS_CONSUMER_<peripheral_name>_<signal_name>. For example, the PRS_CONSUMER_TIMER0_CC0 register is used to select which PRS channel output is sent to the TIMER0 peripheral's CC0 signal. In turn, the target peripheral should be configured to use the associated PRS trigger as desired. This is described in the individual peripheral chapters.

Note: When configuring the synchronous PRS consumer registers, the target peripheral should be disabled or configured to not use the affected PRS signal. This will ensure that no false triggers occur at the consumer.

14.3.4.1 Event on PRS

The PRS can be used to send events to the MCU to wake the system. This is very useful in combination with the Wait For Event (WFE) instruction. Any asynchronous PRS channel can be selected for this using PRSSEL in PRS_CONSUMER_CORE_M33RXEV.

Using this feature, one can e.g. set up a timer to trigger an event to the MCU periodically, every time letting the MCU continue from a WFE instruction in its program. This can help in performance-critical sections where timing is known, and the goal is to wait for an event, execute some code, then wait for another event, execute some code, and so on.

14.3.4.2 DMA Request on PRS

Up to two independent DMA requests can be generated by the PRS. The PRS asynchronous channels triggering the DMA requests are selected with the PRSSEL fields in the PRS_CONSUMER_LDMAXBARn_DMAREQx registers.The requests are set whenever the selected asynchronous PRS outputs are high.

14.4 PRS Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	PRS_IPVERSION	R	PRS IPVERSION
0x008	PRS_ASYNC_SWPULSE	W	Software Pulse Register
0x00C	PRS_ASYNC_SWLEVEL	RW	Software Level Register
0x010	PRS_ASYNC_PEEK	RH	Async Channel Values
0x014	PRS_SYNC_PEEK	RH	Sync Channel Values
0x018	PRS_ASYNC_CHx_CTRL	RW	Async Channel Control Register
0x058	PRS_SYNC_CHx_CTRL	RW	Sync Channel Control Register
0x068	PRS_CONSUMER_EUSART0_CLK	RW	CLK Consumer Register
0x06C	PRS_CONSUMER_EUSART0_RX	RW	RX Consumer Register
0x070	PRS_CONSUMER_EUSART0_TRIGGER	RW	TRIGGER Consumer Register
0x074	PRS_CONSUMER_EUSART1_CLK	RW	CLK Consumer Register
0x078	PRS_CONSUMER_EUSART1_RX	RW	RX Consumer Register
0x07C	PRS_CONSUMER_EUSART1_TRIGGER	RW	TRIGGER Consumer Register
0x080	PRS_CONSUMER_EUSART2_CLK	RW	CLK Consumer Register
0x084	PRS_CONSUMER_EUSART2_RX	RW	RX Consumer Register
0x088	PRS_CONSUMER_EUSART2_TRIGGER	RW	TRIGGER Consumer Register
0x08C	PRS_CONSUMER_FRC_RXRAW	RW	RXRAW Consumer Register
0x090	PRS_CONSUMER_CMU_CALDN	RW	CALDN Consumer Register
0x094	PRS_CONSUMER_CMU_CALUP	RW	CALUP Consumer Register
0x098	PRS_CONSUMER_LDMAXBAR0_DMAREQ0	RW	DMAREQ0 Consumer Register
0x09C	PRS_CONSUMER_LDMAXBAR0_DMAREQ1	RW	DMAREQ1 Consumer Register
0x0A0	PRS_CONSUMER_LEDDRV0_LEDDRV0ASYNC	RW	LEDDRV0 Consumer Register
0x0A4	PRS_CONSUMER_LEDDRV0_LEDDRV1ASYNC	RW	LEDDRV1 Consumer Register
0x0A8	PRS_CONSUMER_LETIMER0_CLEAR	RW	CLEAR Consumer Register
0x0AC	PRS_CONSUMER_LETIMER0_START	RW	START Consumer Register

Offset	Name	Type	Description
0x0B0	PRS_CONSUMER_LETIMER0_STOP	RW	STOP Consumer Register
0x0B4	PRS_CONSUMER_MODEM_DIN	RW	DIN Consumer Register
0x0B8	PRS_CONSUMER_MODEM_PAEN	RW	PAEN Consumer Register
0x0BC	PRS_CONSUMER_PCNT0_S0IN	RW	S0IN Consumer Register
0x0C0	PRS_CONSUMER_PCNT0_S1IN	RW	S1IN Consumer Register
0x0C4	PRS_CONSUMER_PIXELRZ0_INASYNC	RW	IN Consumer Register
0x0C8	PRS_CONSUMER_PIXELRZ1_INASYNC	RW	IN Consumer Register
0x0F8	PRS_CONSUMER_RAC_CLR	RW	CLR Consumer Register
0x0FC	PRS_CONSUMER_RAC_CTIIN0	RW	CTI Consumer Register
0x100	PRS_CONSUMER_RAC_CTIIN1	RW	CTI Consumer Register
0x104	PRS_CONSUMER_RAC_CTIIN2	RW	CTI Consumer Register
0x108	PRS_CONSUMER_RAC_CTIIN3	RW	CTI Consumer Register
0x10C	PRS_CONSUMER_RAC_FORCETX	RW	FORCETX Consumer Register
0x110	PRS_CONSUMER_RAC_RXDIS	RW	RXDIS Consumer Register
0x114	PRS_CONSUMER_RAC_RXEN	RW	RXEN Consumer Register
0x118	PRS_CONSUMER_RAC_TXEN	RW	TXEN Consumer Register
0x11C	PRS_CONSUMER_RFTIMER_PULSE	RW	PULSE Consumer Register
0x120	PRS_CONSUMER_ADC0_ASYNCSEQ	RW	ASYNC Consumer Register
0x124	PRS_CONSUMER_ADC0_SYNCSEQ	RW	SYNC Consumer Register
0x128	PRS_CONSUMER_SEQACC_WAIT0	RW	WAIT0 Consumer Register
0x12C	PRS_CONSUMER_SEQACC_WAIT1	RW	WAIT1 Consumer Register
0x130	PRS_CONSUMER_SEQACC_WAIT2	RW	WAIT2 Consumer Register
0x134	PRS_CONSUMER_SEQACC_WAIT3	RW	WAIT3 Consumer Register
0x138	PRS_CONSUMER_SETAMPER_TAMPERSRC29	RW	TAMPERSRC29 Consumer Register
0x13C	PRS_CONSUMER_SETAMPER_TAMPERSRC30	RW	TAMPERSRC30 Consumer Register
0x140	PRS_CONSUMER_SETAMPER_TAMPERSRC31	RW	TAMPERSRC31 Consumer Register
0x144	PRS_CONSUMER_SYSRTC0_IN0	RW	IN0 Consumer Register
0x148	PRS_CONSUMER_SYSRTC0_IN1	RW	IN1 Consumer Register

Offset	Name	Type	Description
0x14C	PRS_CONSUMER_HFXO0_OSCREQ	RW	OSC Consumer Register
0x150	PRS_CONSUMER_HFXO0_TIMEOUT	RW	TIMEOUT Consumer Register
0x154	PRS_CONSUMER_CORE_CTIIN0	RW	CTI Consumer Register
0x158	PRS_CONSUMER_CORE_CTIIN1	RW	CTI Consumer Register
0x15C	PRS_CONSUMER_CORE_CTIIN2	RW	CTI Consumer Register
0x160	PRS_CONSUMER_CORE_CTIIN3	RW	CTI Consumer Register
0x164	PRS_CONSUMER_CORE_M33RXEV	RW	M33 Consumer Register
0x168	PRS_CONSUMER_TIMER0_CC0	RW	CC0 Consumer Register
0x16C	PRS_CONSUMER_TIMER0_CC1	RW	CC1 Consumer Register
0x170	PRS_CONSUMER_TIMER0_CC2	RW	CC2 Consumer Register
0x174	PRS_CONSUMER_TIMER0_DTI	RW	DTI Consumer Register
0x178	PRS_CONSUMER_TIMER0_DTIFS1	RW	DTI Consumer Register
0x17C	PRS_CONSUMER_TIMER0_DTIFS2	RW	DTI Consumer Register
0x18C	PRS_CONSUMER_TIMER1_CC0	RW	CC0 Consumer Register
0x190	PRS_CONSUMER_TIMER1_CC1	RW	CC1 Consumer Register
0x194	PRS_CONSUMER_TIMER1_CC2	RW	CC2 Consumer Register
0x198	PRS_CONSUMER_TIMER1_DTI	RW	DTI Consumer Register
0x19C	PRS_CONSUMER_TIMER1_DTIFS1	RW	DTI Consumer Register
0x1A0	PRS_CONSUMER_TIMER1_DTIFS2	RW	DTI Consumer Register
0x1A4	PRS_CONSUMER_TIMER2_CC0	RW	CC0 Consumer Register
0x1A8	PRS_CONSUMER_TIMER2_CC1	RW	CC1 Consumer Register
0x1AC	PRS_CONSUMER_TIMER2_CC2	RW	CC2 Consumer Register
0x1B0	PRS_CONSUMER_TIMER2_CC3	RW	CC3 Consumer Register
0x1B4	PRS_CONSUMER_TIMER2_CC4	RW	CC4 Consumer Register
0x1B8	PRS_CONSUMER_TIMER2_CC5	RW	CC5 Consumer Register
0x1BC	PRS_CONSUMER_TIMER2_CC6	RW	CC6 Consumer Register

Offset	Name	Type	Description
0x1C0	PRS_CONSUMER_TIMER2_DTI	RW	DTI Consumer Register
0x1C4	PRS_CONSUMER_TIMER2_DTIFS1	RW	DTI Consumer Register
0x1C8	PRS_CONSUMER_TIMER2_DTIFS2	RW	DTI Consumer Register
0x1CC	PRS_CONSUMER_TIMER3_CC0	RW	CC0 Consumer Register
0x1D0	PRS_CONSUMER_TIMER3_CC1	RW	CC1 Consumer Register
0x1D4	PRS_CONSUMER_TIMER3_CC2	RW	CC2 Consumer Register
0x1D8	PRS_CONSUMER_TIMER3_CC3	RW	CC3 Consumer Register
0x1DC	PRS_CONSUMER_TIMER3_CC4	RW	CC4 Consumer Register
0x1E0	PRS_CONSUMER_TIMER3_CC5	RW	CC5 Consumer Register
0x1E4	PRS_CONSUMER_TIMER3_CC6	RW	CC6 Consumer Register
0x1E8	PRS_CONSUMER_TIMER3_DTI	RW	DTI Consumer Register
0x1EC	PRS_CONSUMER_TIMER3_DTIFS1	RW	DTI Consumer Register
0x1F0	PRS_CONSUMER_TIMER3_DTIFS2	RW	DTI Consumer Register
0x1F4	PRS_CONSUMER_WDOG0_SRC0	RW	SRC0 Consumer Register
0x1F8	PRS_CONSUMER_WDOG0_SRC1	RW	SRC1 Consumer Register
0x1FC	PRS_CONSUMER_WDOG1_SRC0	RW	SRC0 Consumer Register
0x200	PRS_CONSUMER_WDOG1_SRC1	RW	SRC1 Consumer Register
0x214	PRS_RPURATD4	RW	Root Access Type Descriptor Register
0x1000	PRS_IPVERSION_SET	R	PRS IPVERSION
0x1008	PRS_ASYNC_SWPULSE_SET	W	Software Pulse Register
0x100C	PRS_ASYNC_SWLEVEL_SET	RW	Software Level Register
0x1010	PRS_ASYNC_PEEK_SET	RH	Async Channel Values
0x1014	PRS_SYNC_PEEK_SET	RH	Sync Channel Values
0x1018	PRS_ASYNC_CHx_CTRL_SET	RW	Async Channel Control Register
0x1058	PRS_SYNC_CHx_CTRL_SET	RW	Sync Channel Control Register
0x1068	PRS_CONSUMER_EUSART0_CLK_SET	RW	CLK Consumer Register
0x106C	PRS_CONSUMER_EUSART0_RX_SET	RW	RX Consumer Register
0x1070	PRS_CONSUMER_EUSART0_TRIGGER_SET	RW	TRIGGER Consumer Register
0x1074	PRS_CONSUMER_EUSART1_CLK_SET	RW	CLK Consumer Register

Offset	Name	Type	Description
0x1078	PRS_CONSUMER_EUSART1_RX_SET	RW	RX Consumer Register
0x107C	PRS_CONSUMER_EUSART1_TRIGGER_SET	RW	TRIGGER Consumer Register
0x1080	PRS_CONSUMER_EUSART2_CLK_SET	RW	CLK Consumer Register
0x1084	PRS_CONSUMER_EUSART2_RX_SET	RW	RX Consumer Register
0x1088	PRS_CONSUMER_EUSART2_TRIGGER_SET	RW	TRIGGER Consumer Register
0x108C	PRS_CONSUMER_FRC_RXRAW_SET	RW	RXRAW Consumer Register
0x1090	PRS_CONSUMER_CMU_CALDN_SET	RW	CALDN Consumer Register
0x1094	PRS_CONSUMER_CMU_CALUP_SET	RW	CALUP Consumer Register
0x1098	PRS_CONSUMER_LDMAXBAR0_DMAREQ0_SET	RW	DMAREQ0 Consumer Register
0x109C	PRS_CONSUMER_LDMAXBAR0_DMAREQ1_SET	RW	DMAREQ1 Consumer Register
0x10A0	PRS_CONSUMER_LEDDRV0_LEDDRV0ASYNC_SET	RW	LEDDRV0 Consumer Register
0x10A4	PRS_CONSUMER_LEDDRV0_LEDDRV1ASYNC_SET	RW	LEDDRV1 Consumer Register
0x10A8	PRS_CONSUMER_LETIMER0_CLEAR_SET	RW	CLEAR Consumer Register
0x10AC	PRS_CONSUMER_LETIMER0_START_SET	RW	START Consumer Register
0x10B0	PRS_CONSUMER_LETIMER0_STOP_SET	RW	STOP Consumer Register
0x10B4	PRS_CONSUMER_MODEM_DIN_SET	RW	DIN Consumer Register
0x10B8	PRS_CONSUMER_MODEM_PAEN_SET	RW	PAEN Consumer Register
0x10BC	PRS_CONSUMER_PCNT0_S0IN_SET	RW	S0IN Consumer Register
0x10C0	PRS_CONSUMER_PCNT0_S1IN_SET	RW	S1IN Consumer Register
0x10C4	PRS_CONSUMER_PIXELRZ0_INASYNC_SET	RW	IN Consumer Register
0x10C8	PRS_CONSUMER_PIXELRZ1_INASYNC_SET	RW	IN Consumer Register
0x10F8	PRS_CONSUMER_RAC_CLR_SET	RW	CLR Consumer Register
0x10FC	PRS_CONSUMER_RAC_CTIIN0_SET	RW	CTI Consumer Register
0x1100	PRS_CONSUMER_RAC_CTIIN1_SET	RW	CTI Consumer Register

Offset	Name	Type	Description
0x1104	PRS_CONSUMER_RAC_CTIIN2_SET	RW	CTI Consumer Register
0x1108	PRS_CONSUMER_RAC_CTIIN3_SET	RW	CTI Consumer Register
0x110C	PRS_CONSUMER_RAC_FORCETX_SET	RW	FORCETX Consumer Register
0x1110	PRS_CONSUMER_RAC_RXDIS_SET	RW	RXDIS Consumer Register
0x1114	PRS_CONSUMER_RAC_RXEN_SET	RW	RXEN Consumer Register
0x1118	PRS_CONSUMER_RAC_TXEN_SET	RW	TXEN Consumer Register
0x111C	PRS_CONSUMER_RFTIMER_PULSE_SET	RW	PULSE Consumer Register
0x1120	PRS_CONSUMER_ADC0_ASYNCSEQ_SET	RW	ASYNC Consumer Register
0x1124	PRS_CONSUMER_ADC0_SYNCSEQ_SET	RW	SYNC Consumer Register
0x1128	PRS_CONSUMER_SEQACC_WAIT0_SET	RW	WAIT0 Consumer Register
0x112C	PRS_CONSUMER_SEQACC_WAIT1_SET	RW	WAIT1 Consumer Register
0x1130	PRS_CONSUMER_SEQACC_WAIT2_SET	RW	WAIT2 Consumer Register
0x1134	PRS_CONSUMER_SEQACC_WAIT3_SET	RW	WAIT3 Consumer Register
0x1138	PRS_CONSUMER_SETAMPER_TAMPERSRC29_SET	RW	TAMPERSRC29 Consumer Register
0x113C	PRS_CONSUMER_SETAMPER_TAMPERSRC30_SET	RW	TAMPERSRC30 Consumer Register
0x1140	PRS_CONSUMER_SETAMPER_TAMPERSRC31_SET	RW	TAMPERSRC31 Consumer Register
0x1144	PRS_CONSUMER_SYSRTC0_IN0_SET	RW	IN0 Consumer Register
0x1148	PRS_CONSUMER_SYSRTC0_IN1_SET	RW	IN1 Consumer Register
0x114C	PRS_CONSUMER_HFXO0_OSCREQ_SET	RW	OSC Consumer Register
0x1150	PRS_CONSUMER_HFXO0_TIMEOUT_SET	RW	TIMEOUT Consumer Register
0x1154	PRS_CONSUMER_CORE_CTIIN0_SET	RW	CTI Consumer Register
0x1158	PRS_CONSUMER_CORE_CTIIN1_SET	RW	CTI Consumer Register
0x115C	PRS_CONSUMER_CORE_CTIIN2_SET	RW	CTI Consumer Register
0x1160	PRS_CONSUMER_CORE_CTIIN3_SET	RW	CTI Consumer Register
0x1164	PRS_CONSUMER_CORE_M33RXEV_SET	RW	M33 Consumer Register

Offset	Name	Type	Description
0x1168	PRS_CONSUMER_TIMER0_CC0_SET	RW	CC0 Consumer Register
0x116C	PRS_CONSUMER_TIMER0_CC1_SET	RW	CC1 Consumer Register
0x1170	PRS_CONSUMER_TIMER0_CC2_SET	RW	CC2 Consumer Register
0x1174	PRS_CONSUMER_TIMER0_DTI_SET	RW	DTI Consumer Register
0x1178	PRS_CONSUMER_TIMER0_DTIFS1_SET	RW	DTI Consumer Register
0x117C	PRS_CONSUMER_TIMER0_DTIFS2_SET	RW	DTI Consumer Register
0x118C	PRS_CONSUMER_TIMER1_CC0_SET	RW	CC0 Consumer Register
0x1190	PRS_CONSUMER_TIMER1_CC1_SET	RW	CC1 Consumer Register
0x1194	PRS_CONSUMER_TIMER1_CC2_SET	RW	CC2 Consumer Register
0x1198	PRS_CONSUMER_TIMER1_DTI_SET	RW	DTI Consumer Register
0x119C	PRS_CONSUMER_TIMER1_DTIFS1_SET	RW	DTI Consumer Register
0x11A0	PRS_CONSUMER_TIMER1_DTIFS2_SET	RW	DTI Consumer Register
0x11A4	PRS_CONSUMER_TIMER2_CC0_SET	RW	CC0 Consumer Register
0x11A8	PRS_CONSUMER_TIMER2_CC1_SET	RW	CC1 Consumer Register
0x11AC	PRS_CONSUMER_TIMER2_CC2_SET	RW	CC2 Consumer Register
0x11B0	PRS_CONSUMER_TIMER2_CC3_SET	RW	CC3 Consumer Register
0x11B4	PRS_CONSUMER_TIMER2_CC4_SET	RW	CC4 Consumer Register
0x11B8	PRS_CONSUMER_TIMER2_CC5_SET	RW	CC5 Consumer Register
0x11BC	PRS_CONSUMER_TIMER2_CC6_SET	RW	CC6 Consumer Register
0x11C0	PRS_CONSUMER_TIMER2_DTI_SET	RW	DTI Consumer Register
0x11C4	PRS_CONSUMER_TIMER2_DTIFS1_SET	RW	DTI Consumer Register
0x11C8	PRS_CONSUMER_TIMER2_DTIFS2_SET	RW	DTI Consumer Register
0x11CC	PRS_CONSUMER_TIMER3_CC0_SET	RW	CC0 Consumer Register
0x11D0	PRS_CONSUMER_TIMER3_CC1_SET	RW	CC1 Consumer Register
0x11D4	PRS_CONSUMER_TIMER3_CC2_SET	RW	CC2 Consumer Register

Offset	Name	Type	Description
0x11D8	PRS_CONSUMER_TIMER3_CC3_SET	RW	CC3 Consumer Register
0x11DC	PRS_CONSUMER_TIMER3_CC4_SET	RW	CC4 Consumer Register
0x11E0	PRS_CONSUMER_TIMER3_CC5_SET	RW	CC5 Consumer Register
0x11E4	PRS_CONSUMER_TIMER3_CC6_SET	RW	CC6 Consumer Register
0x11E8	PRS_CONSUMER_TIMER3_DTI_SET	RW	DTI Consumer Register
0x11EC	PRS_CONSUMER_TIMER3_DTIFS1_SET	RW	DTI Consumer Register
0x11F0	PRS_CONSUMER_TIMER3_DTIFS2_SET	RW	DTI Consumer Register
0x11F4	PRS_CONSUMER_WDOG0_SRC0_SET	RW	SRC0 Consumer Register
0x11F8	PRS_CONSUMER_WDOG0_SRC1_SET	RW	SRC1 Consumer Register
0x11FC	PRS_CONSUMER_WDOG1_SRC0_SET	RW	SRC0 Consumer Register
0x1200	PRS_CONSUMER_WDOG1_SRC1_SET	RW	SRC1 Consumer Register
0x1214	PRS_RPURATD4_SET	RW	Root Access Type Descriptor Register
0x2000	PRS_IPVERSION_CLR	R	PRS IPVERSION
0x2008	PRS_ASYNC_SWPULSE_CLR	W	Software Pulse Register
0x200C	PRS_ASYNC_SWLEVEL_CLR	RW	Software Level Register
0x2010	PRS_ASYNC_PEEK_CLR	RH	Async Channel Values
0x2014	PRS_SYNC_PEEK_CLR	RH	Sync Channel Values
0x2018	PRS_ASYNC_CHx_CTRL_CLR	RW	Async Channel Control Register
0x2058	PRS_SYNC_CHx_CTRL_CLR	RW	Sync Channel Control Register
0x2068	PRS_CONSUMER_EUSART0_CLK_CLR	RW	CLK Consumer Register
0x206C	PRS_CONSUMER_EUSART0_RX_CLR	RW	RX Consumer Register
0x2070	PRS_CONSUMER_EUSART0_TRIGGER_CLR	RW	TRIGGER Consumer Register
0x2074	PRS_CONSUMER_EUSART1_CLK_CLR	RW	CLK Consumer Register
0x2078	PRS_CONSUMER_EUSART1_RX_CLR	RW	RX Consumer Register
0x207C	PRS_CONSUMER_EUSART1_TRIGGER_CLR	RW	TRIGGER Consumer Register
0x2080	PRS_CONSUMER_EUSART2_CLK_CLR	RW	CLK Consumer Register
0x2084	PRS_CONSUMER_EUSART2_RX_CLR	RW	RX Consumer Register
0x2088	PRS_CONSUMER_EUSART2_TRIGGER_CLR	RW	TRIGGER Consumer Register

Offset	Name	Type	Description
0x208C	PRS_CONSUMER_FRC_RXRAW_CLR	RW	RXRAW Consumer Register
0x2090	PRS_CONSUMER_CMU_CALDN_CLR	RW	CALDN Consumer Register
0x2094	PRS_CONSUMER_CMU_CALUP_CLR	RW	CALUP Consumer Register
0x2098	PRS_CONSUMER_LDMAXBAR0_DMAREQ0_CLR	RW	DMAREQ0 Consumer Register
0x209C	PRS_CONSUMER_LDMAXBAR0_DMAREQ1_CLR	RW	DMAREQ1 Consumer Register
0x20A0	PRS_CONSUMER_LEDDRV0_LEDDRV0ASYNC_CLR	RW	LEDDRV0 Consumer Register
0x20A4	PRS_CONSUMER_LEDDRV0_LEDDRV1ASYNC_CLR	RW	LEDDRV1 Consumer Register
0x20A8	PRS_CONSUMER_LETIMER0_CLEAR_CLR	RW	CLEAR Consumer Register
0x20AC	PRS_CONSUMER_LETIMER0_START_CLR	RW	START Consumer Register
0x20B0	PRS_CONSUMER_LETIMER0_STOP_CLR	RW	STOP Consumer Register
0x20B4	PRS_CONSUMER_MODEM_DIN_CLR	RW	DIN Consumer Register
0x20B8	PRS_CONSUMER_MODEM_PAEN_CLR	RW	PAEN Consumer Register
0x20BC	PRS_CONSUMER_PCNT0_S0IN_CLR	RW	S0IN Consumer Register
0x20C0	PRS_CONSUMER_PCNT0_S1IN_CLR	RW	S1IN Consumer Register
0x20C4	PRS_CONSUMER_PIXELRZ0_INASYNC_CLR	RW	IN Consumer Register
0x20C8	PRS_CONSUMER_PIXELRZ1_INASYNC_CLR	RW	IN Consumer Register
0x20F8	PRS_CONSUMER_RAC_CLR_CLR	RW	CLR Consumer Register
0x20FC	PRS_CONSUMER_RAC_CTIIN0_CLR	RW	CTI Consumer Register
0x2100	PRS_CONSUMER_RAC_CTIIN1_CLR	RW	CTI Consumer Register
0x2104	PRS_CONSUMER_RAC_CTIIN2_CLR	RW	CTI Consumer Register
0x2108	PRS_CONSUMER_RAC_CTIIN3_CLR	RW	CTI Consumer Register
0x210C	PRS_CONSUMER_RAC_FORCETX_CLR	RW	FORCETX Consumer Register
0x2110	PRS_CONSUMER_RAC_RXDIS_CLR	RW	RXDIS Consumer Register
0x2114	PRS_CONSUMER_RAC_RXEN_CLR	RW	RXEN Consumer Register

Offset	Name	Type	Description
0x2118	PRS_CONSUMER_RAC_TXEN_CLR	RW	TXEN Consumer Register
0x211C	PRS_CONSUMER_RFTIMER_PULSE_CLR	RW	PULSE Consumer Register
0x2120	PRS_CONSUMER_ADC0_ASYNCSEQ_CLR	RW	ASYNC Consumer Register
0x2124	PRS_CONSUMER_ADC0_SYNCSEQ_CLR	RW	SYNC Consumer Register
0x2128	PRS_CONSUMER_SEQACC_WAIT0_CLR	RW	WAIT0 Consumer Register
0x212C	PRS_CONSUMER_SEQACC_WAIT1_CLR	RW	WAIT1 Consumer Register
0x2130	PRS_CONSUMER_SEQACC_WAIT2_CLR	RW	WAIT2 Consumer Register
0x2134	PRS_CONSUMER_SEQACC_WAIT3_CLR	RW	WAIT3 Consumer Register
0x2138	PRS_CONSUMER_SETAMPER_TAMPERSRC29_CLR	RW	TAMPERSRC29 Consumer Register
0x213C	PRS_CONSUMER_SETAMPER_TAMPERSRC30_CLR	RW	TAMPERSRC30 Consumer Register
0x2140	PRS_CONSUMER_SETAMPER_TAMPERSRC31_CLR	RW	TAMPERSRC31 Consumer Register
0x2144	PRS_CONSUMER_SYSRTC0_IN0_CLR	RW	IN0 Consumer Register
0x2148	PRS_CONSUMER_SYSRTC0_IN1_CLR	RW	IN1 Consumer Register
0x214C	PRS_CONSUMER_HFXO0_OSCREQ_CLR	RW	OSC Consumer Register
0x2150	PRS_CONSUMER_HFXO0_TIMEOUT_CLR	RW	TIMEOUT Consumer Register
0x2154	PRS_CONSUMER_CORE_CTIIN0_CLR	RW	CTI Consumer Register
0x2158	PRS_CONSUMER_CORE_CTIIN1_CLR	RW	CTI Consumer Register
0x215C	PRS_CONSUMER_CORE_CTIIN2_CLR	RW	CTI Consumer Register
0x2160	PRS_CONSUMER_CORE_CTIIN3_CLR	RW	CTI Consumer Register
0x2164	PRS_CONSUMER_CORE_M33RXEV_CLR	RW	M33 Consumer Register
0x2168	PRS_CONSUMER_TIMER0_CC0_CLR	RW	CC0 Consumer Register
0x216C	PRS_CONSUMER_TIMER0_CC1_CLR	RW	CC1 Consumer Register
0x2170	PRS_CONSUMER_TIMER0_CC2_CLR	RW	CC2 Consumer Register
0x2174	PRS_CONSUMER_TIMER0_DTI_CLR	RW	DTI Consumer Register
0x2178	PRS_CONSUMER_TIMER0_DTIFS1_CLR	RW	DTI Consumer Register

Offset	Name	Type	Description
0x217C	PRS_CONSUMER_TIMER0_DTIFS2_CLR	RW	DTI Consumer Register
0x218C	PRS_CONSUMER_TIMER1_CC0_CLR	RW	CC0 Consumer Register
0x2190	PRS_CONSUMER_TIMER1_CC1_CLR	RW	CC1 Consumer Register
0x2194	PRS_CONSUMER_TIMER1_CC2_CLR	RW	CC2 Consumer Register
0x2198	PRS_CONSUMER_TIMER1_DTI_CLR	RW	DTI Consumer Register
0x219C	PRS_CONSUMER_TIMER1_DTIFS1_CLR	RW	DTI Consumer Register
0x21A0	PRS_CONSUMER_TIMER1_DTIFS2_CLR	RW	DTI Consumer Register
0x21A4	PRS_CONSUMER_TIMER2_CC0_CLR	RW	CC0 Consumer Register
0x21A8	PRS_CONSUMER_TIMER2_CC1_CLR	RW	CC1 Consumer Register
0x21AC	PRS_CONSUMER_TIMER2_CC2_CLR	RW	CC2 Consumer Register
0x21B0	PRS_CONSUMER_TIMER2_CC3_CLR	RW	CC3 Consumer Register
0x21B4	PRS_CONSUMER_TIMER2_CC4_CLR	RW	CC4 Consumer Register
0x21B8	PRS_CONSUMER_TIMER2_CC5_CLR	RW	CC5 Consumer Register
0x21BC	PRS_CONSUMER_TIMER2_CC6_CLR	RW	CC6 Consumer Register
0x21C0	PRS_CONSUMER_TIMER2_DTI_CLR	RW	DTI Consumer Register
0x21C4	PRS_CONSUMER_TIMER2_DTIFS1_CLR	RW	DTI Consumer Register
0x21C8	PRS_CONSUMER_TIMER2_DTIFS2_CLR	RW	DTI Consumer Register
0x21CC	PRS_CONSUMER_TIMER3_CC0_CLR	RW	CC0 Consumer Register
0x21D0	PRS_CONSUMER_TIMER3_CC1_CLR	RW	CC1 Consumer Register
0x21D4	PRS_CONSUMER_TIMER3_CC2_CLR	RW	CC2 Consumer Register
0x21D8	PRS_CONSUMER_TIMER3_CC3_CLR	RW	CC3 Consumer Register
0x21DC	PRS_CONSUMER_TIMER3_CC4_CLR	RW	CC4 Consumer Register
0x21E0	PRS_CONSUMER_TIMER3_CC5_CLR	RW	CC5 Consumer Register
0x21E4	PRS_CONSUMER_TIMER3_CC6_CLR	RW	CC6 Consumer Register
0x21E8	PRS_CONSUMER_TIMER3_DTI_CLR	RW	DTI Consumer Register

Offset	Name	Type	Description
0x21EC	PRS_CONSUMER_TIMER3_DTIFS1_CLR	RW	DTI Consumer Register
0x21F0	PRS_CONSUMER_TIMER3_DTIFS2_CLR	RW	DTI Consumer Register
0x21F4	PRS_CONSUMER_WDOG0_SRC0_CLR	RW	SRC0 Consumer Register
0x21F8	PRS_CONSUMER_WDOG0_SRC1_CLR	RW	SRC1 Consumer Register
0x21FC	PRS_CONSUMER_WDOG1_SRC0_CLR	RW	SRC0 Consumer Register
0x2200	PRS_CONSUMER_WDOG1_SRC1_CLR	RW	SRC1 Consumer Register
0x2214	PRS_RPURATD4_CLR	RW	Root Access Type Descriptor Register
0x3000	PRS_IPVERSION_TGL	R	PRS IPVERSION
0x3008	PRS_ASYNC_SWPULSE_TGL	W	Software Pulse Register
0x300C	PRS_ASYNC_SWLEVEL_TGL	RW	Software Level Register
0x3010	PRS_ASYNC_PEEK_TGL	RH	Async Channel Values
0x3014	PRS_SYNC_PEEK_TGL	RH	Sync Channel Values
0x3018	PRS_ASYNC_CHx_CTRL_TGL	RW	Async Channel Control Register
0x3058	PRS_SYNC_CHx_CTRL_TGL	RW	Sync Channel Control Register
0x3068	PRS_CONSUMER_EUSART0_CLK_TGL	RW	CLK Consumer Register
0x306C	PRS_CONSUMER_EUSART0_RX_TGL	RW	RX Consumer Register
0x3070	PRS_CONSUMER_EUSART0_TRIGGER_TGL	RW	TRIGGER Consumer Register
0x3074	PRS_CONSUMER_EUSART1_CLK_TGL	RW	CLK Consumer Register
0x3078	PRS_CONSUMER_EUSART1_RX_TGL	RW	RX Consumer Register
0x307C	PRS_CONSUMER_EUSART1_TRIGGER_TGL	RW	TRIGGER Consumer Register
0x3080	PRS_CONSUMER_EUSART2_CLK_TGL	RW	CLK Consumer Register
0x3084	PRS_CONSUMER_EUSART2_RX_TGL	RW	RX Consumer Register
0x3088	PRS_CONSUMER_EUSART2_TRIGGER_TGL	RW	TRIGGER Consumer Register
0x308C	PRS_CONSUMER_FRC_RXRAW_TGL	RW	RXRAW Consumer Register
0x3090	PRS_CONSUMER_CMU_CALDN_TGL	RW	CALDN Consumer Register
0x3094	PRS_CONSUMER_CMU_CALUP_TGL	RW	CALUP Consumer Register
0x3098	PRS_CONSUMER_LDMAXBAR0_DMAREQ0_TGL	RW	DMAREQ0 Consumer Register
0x309C	PRS_CONSUMER_LDMAXBAR0_DMAREQ1_TGL	RW	DMAREQ1 Consumer Register

Offset	Name	Type	Description
0x30A0	PRS_CONSUMER_LEDDRV0_LEDDRV0ASYNC_TGL	RW	LEDDRV0 Consumer Register
0x30A4	PRS_CONSUMER_LEDDRV0_LEDDRV1ASYNC_TGL	RW	LEDDRV1 Consumer Register
0x30A8	PRS_CONSUMER_LETIMER0_CLEAR_TGL	RW	CLEAR Consumer Register
0x30AC	PRS_CONSUMER_LETIMER0_START_TGL	RW	START Consumer Register
0x30B0	PRS_CONSUMER_LETIMER0_STOP_TGL	RW	STOP Consumer Register
0x30B4	PRS_CONSUMER_MODEM_DIN_TGL	RW	DIN Consumer Register
0x30B8	PRS_CONSUMER_MODEM_PAEN_TGL	RW	PAEN Consumer Register
0x30BC	PRS_CONSUMER_PCNT0_S0IN_TGL	RW	S0IN Consumer Register
0x30C0	PRS_CONSUMER_PCNT0_S1IN_TGL	RW	S1IN Consumer Register
0x30C4	PRS_CONSUMER_PIXELRZ0_INASYNC_TGL	RW	IN Consumer Register
0x30C8	PRS_CONSUMER_PIXELRZ1_INASYNC_TGL	RW	IN Consumer Register
0x30F8	PRS_CONSUMER_RAC_CLR_TGL	RW	CLR Consumer Register
0x30FC	PRS_CONSUMER_RAC_CTIIN0_TGL	RW	CTI Consumer Register
0x3100	PRS_CONSUMER_RAC_CTIIN1_TGL	RW	CTI Consumer Register
0x3104	PRS_CONSUMER_RAC_CTIIN2_TGL	RW	CTI Consumer Register
0x3108	PRS_CONSUMER_RAC_CTIIN3_TGL	RW	CTI Consumer Register
0x310C	PRS_CONSUMER_RAC_FORCETX_TGL	RW	FORCETX Consumer Register
0x3110	PRS_CONSUMER_RAC_RXDIS_TGL	RW	RXDIS Consumer Register
0x3114	PRS_CONSUMER_RAC_RXEN_TGL	RW	RXEN Consumer Register
0x3118	PRS_CONSUMER_RAC_TXEN_TGL	RW	TXEN Consumer Register
0x311C	PRS_CONSUMER_RFTIMER_PULSE_TGL	RW	PULSE Consumer Register
0x3120	PRS_CONSUMER_ADC0_ASYNCSEQ_TGL	RW	ASYNC Consumer Register
0x3124	PRS_CONSUMER_ADC0_SYNCSEQ_TGL	RW	SYNC Consumer Register
0x3128	PRS_CONSUMER_SEQACC_WAIT0_TGL	RW	WAIT0 Consumer Register

Offset	Name	Type	Description
0x312C	PRS_CONSUMER_SEQACC_WAIT1_TGL	RW	WAIT1 Consumer Register
0x3130	PRS_CONSUMER_SEQACC_WAIT2_TGL	RW	WAIT2 Consumer Register
0x3134	PRS_CONSUMER_SEQACC_WAIT3_TGL	RW	WAIT3 Consumer Register
0x3138	PRS_CONSUMER_SETAMPER_TAMPERSRC29_TGL	RW	TAMPERSRC29 Consumer Register
0x313C	PRS_CONSUMER_SETAMPER_TAMPERSRC30_TGL	RW	TAMPERSRC30 Consumer Register
0x3140	PRS_CONSUMER_SETAMPER_TAMPERSRC31_TGL	RW	TAMPERSRC31 Consumer Register
0x3144	PRS_CONSUMER_SYSRTC0_IN0_TGL	RW	IN0 Consumer Register
0x3148	PRS_CONSUMER_SYSRTC0_IN1_TGL	RW	IN1 Consumer Register
0x314C	PRS_CONSUMER_HFXO0_OSCREQ_TGL	RW	OSC Consumer Register
0x3150	PRS_CONSUMER_HFXO0_TIMEOUT_TGL	RW	TIMEOUT Consumer Register
0x3154	PRS_CONSUMER_CORE_CTIIN0_TGL	RW	CTI Consumer Register
0x3158	PRS_CONSUMER_CORE_CTIIN1_TGL	RW	CTI Consumer Register
0x315C	PRS_CONSUMER_CORE_CTIIN2_TGL	RW	CTI Consumer Register
0x3160	PRS_CONSUMER_CORE_CTIIN3_TGL	RW	CTI Consumer Register
0x3164	PRS_CONSUMER_CORE_M33RXEV_TGL	RW	M33 Consumer Register
0x3168	PRS_CONSUMER_TIMER0_CC0_TGL	RW	CC0 Consumer Register
0x316C	PRS_CONSUMER_TIMER0_CC1_TGL	RW	CC1 Consumer Register
0x3170	PRS_CONSUMER_TIMER0_CC2_TGL	RW	CC2 Consumer Register
0x3174	PRS_CONSUMER_TIMER0_DTI_TGL	RW	DTI Consumer Register
0x3178	PRS_CONSUMER_TIMER0_DTIFS1_TGL	RW	DTI Consumer Register
0x317C	PRS_CONSUMER_TIMER0_DTIFS2_TGL	RW	DTI Consumer Register
0x318C	PRS_CONSUMER_TIMER1_CC0_TGL	RW	CC0 Consumer Register
0x3190	PRS_CONSUMER_TIMER1_CC1_TGL	RW	CC1 Consumer Register
0x3194	PRS_CONSUMER_TIMER1_CC2_TGL	RW	CC2 Consumer Register
0x3198	PRS_CONSUMER_TIMER1_DTI_TGL	RW	DTI Consumer Register

Offset	Name	Type	Description
0x319C	PRS_CONSUMER_TIMER1_DTIFS1_TGL	RW	DTI Consumer Register
0x31A0	PRS_CONSUMER_TIMER1_DTIFS2_TGL	RW	DTI Consumer Register
0x31A4	PRS_CONSUMER_TIMER2_CC0_TGL	RW	CC0 Consumer Register
0x31A8	PRS_CONSUMER_TIMER2_CC1_TGL	RW	CC1 Consumer Register
0x31AC	PRS_CONSUMER_TIMER2_CC2_TGL	RW	CC2 Consumer Register
0x31B0	PRS_CONSUMER_TIMER2_CC3_TGL	RW	CC3 Consumer Register
0x31B4	PRS_CONSUMER_TIMER2_CC4_TGL	RW	CC4 Consumer Register
0x31B8	PRS_CONSUMER_TIMER2_CC5_TGL	RW	CC5 Consumer Register
0x31BC	PRS_CONSUMER_TIMER2_CC6_TGL	RW	CC6 Consumer Register
0x31C0	PRS_CONSUMER_TIMER2_DTI_TGL	RW	DTI Consumer Register
0x31C4	PRS_CONSUMER_TIMER2_DTIFS1_TGL	RW	DTI Consumer Register
0x31C8	PRS_CONSUMER_TIMER2_DTIFS2_TGL	RW	DTI Consumer Register
0x31CC	PRS_CONSUMER_TIMER3_CC0_TGL	RW	CC0 Consumer Register
0x31D0	PRS_CONSUMER_TIMER3_CC1_TGL	RW	CC1 Consumer Register
0x31D4	PRS_CONSUMER_TIMER3_CC2_TGL	RW	CC2 Consumer Register
0x31D8	PRS_CONSUMER_TIMER3_CC3_TGL	RW	CC3 Consumer Register
0x31DC	PRS_CONSUMER_TIMER3_CC4_TGL	RW	CC4 Consumer Register
0x31E0	PRS_CONSUMER_TIMER3_CC5_TGL	RW	CC5 Consumer Register
0x31E4	PRS_CONSUMER_TIMER3_CC6_TGL	RW	CC6 Consumer Register
0x31E8	PRS_CONSUMER_TIMER3_DTI_TGL	RW	DTI Consumer Register
0x31EC	PRS_CONSUMER_TIMER3_DTIFS1_TGL	RW	DTI Consumer Register
0x31F0	PRS_CONSUMER_TIMER3_DTIFS2_TGL	RW	DTI Consumer Register
0x31F4	PRS_CONSUMER_WDOG0_SRC0_TGL	RW	SRC0 Consumer Register
0x31F8	PRS_CONSUMER_WDOG0_SRC1_TGL	RW	SRC1 Consumer Register
0x31FC	PRS_CONSUMER_WDOG1_SRC0_TGL	RW	SRC0 Consumer Register

Offset	Name	Type	Description
0x3200	PRS_CONSUM-ER_WDOG1_SRC1_TGL	RW	SRC1 Consumer Register
0x3214	PRS_RPURATD4_TGL	RW	Root Access Type Descriptor Register

14.5 PRS Register Description

14.5.1 PRS_IPVERSION - PRS IPVERSION

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x9																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x9	R	**IP version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

14.5.2 PRS_ASYNC_SWPULSE - Software Pulse Register

Offset	Bit Position																											
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																					W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																					CH11PULSE	CH10PULSE	CH9PULSE	CH8PULSE	CH7PULSE	CH6PULSE	CH5PULSE	CH4PULSE	CH3PULSE	CH2PULSE	CH1PULSE	CH0PULSE

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	CH11PULSE	0x0	W(nB)	**Channel pulse**
	Channel 11 pulse generation			
10	CH10PULSE	0x0	W(nB)	**Channel pulse**
	Channel 10 pulse generation			
9	CH9PULSE	0x0	W(nB)	**Channel pulse**
	Channel 9 pulse generation			
8	CH8PULSE	0x0	W(nB)	**Channel pulse**
	Channel 8 pulse generation			
7	CH7PULSE	0x0	W(nB)	**Channel pulse**
	Channel 7 pulse generation			
6	CH6PULSE	0x0	W(nB)	**Channel pulse**
	Channel 6 pulse generation			
5	CH5PULSE	0x0	W(nB)	**Channel pulse**
	Channel 5 pulse generation			
4	CH4PULSE	0x0	W(nB)	**Channel pulse**
	Channel 4 pulse generation			
3	CH3PULSE	0x0	W(nB)	**Channel pulse**
	Channel 3 pulse generation			
2	CH2PULSE	0x0	W(nB)	**Channel pulse**
	Channel 2 pulse generation			
1	CH1PULSE	0x0	W(nB)	**Channel pulse**
	Channel 1 pulse generation			
0	CH0PULSE	0x0	W(nB)	**Channel pulse**
	Channel 0 pulse generation			

14.5.3 PRS_ASYNC_SWLEVEL - Software Level Register

Offset	Bit Position																											
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																					CH11LEVEL	CH10LEVEL	CH9LEVEL	CH8LEVEL	CH7LEVEL	CH6LEVEL	CH5LEVEL	CH4LEVEL	CH3LEVEL	CH2LEVEL	CH1LEVEL	CH0LEVEL

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	CH11LEVEL	0x0	RW	**Channel Level**
	Channel 11 Software Level			
10	CH10LEVEL	0x0	RW	**Channel Level**
	Channel 10 Software Level			
9	CH9LEVEL	0x0	RW	**Channel Level**
	Channel 9 Software Level			
8	CH8LEVEL	0x0	RW	**Channel Level**
	Channel 8 Software Level			
7	CH7LEVEL	0x0	RW	**Channel Level**
	Channel 7 Software Level			
6	CH6LEVEL	0x0	RW	**Channel Level**
	Channel 6 Software Level			
5	CH5LEVEL	0x0	RW	**Channel Level**
	Channel 5 Software Level			
4	CH4LEVEL	0x0	RW	**Channel Level**
	Channel 4 Software Level			
3	CH3LEVEL	0x0	RW	**Channel Level**
	Channel 3 Software Level			
2	CH2LEVEL	0x0	RW	**Channel Level**
	Channel 2 Software Level			
1	CH1LEVEL	0x0	RW	**Channel Level**
	Channel 1 Software Level			
0	CH0LEVEL	0x0	RW	**Channel Level**
	Channel 0 Software Level			

14.5.4 PRS_ASYNC_PEEK - Async Channel Values

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																					R	R	R	R	R	R	R	R	R	R	R	R
Name																					CH11VAL	CH10VAL	CH9VAL	CH8VAL	CH7VAL	CH6VAL	CH5VAL	CH4VAL	CH3VAL	CH2VAL	CH1VAL	CH0VAL

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	CH11VAL	0x0	R	**Channel 11 Current Value**
	See bit 0.			
10	CH10VAL	0x0	R	**Channel 10 Current Value**
	See bit 0.			
9	CH9VAL	0x0	R	**Channel 9 Current Value**
	See bit 0.			
8	CH8VAL	0x0	R	**Channel 8 Current Value**
	See bit 0.			
7	CH7VAL	0x0	R	**Channel 7 Current Value**
	See bit 0.			
6	CH6VAL	0x0	R	**Channel 6 Current Value**
	See bit 0.			
5	CH5VAL	0x0	R	**Channel 5 Current Value**
	See bit 0.			
4	CH4VAL	0x0	R	**Channel 4 Current Value**
	See bit 0.			
3	CH3VAL	0x0	R	**Channel 3 Current Value**
	See bit 0.			
2	CH2VAL	0x0	R	**Channel 2 Current Value**
	See bit 0.			
1	CH1VAL	0x0	R	**Channel 1 Current Value**
	See bit 0.			
0	CH0VAL	0x0	R	**Channel 0 Current Value**
	Sample the current output value of channel 0. This value may be one or two clock delayed			

14.5.5 PRS_SYNC_PEEK - Sync Channel Values

Offset	Bit Position																														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													R	R	R	R
Name																													CH3VAL	CH2VAL	CH1VAL	CH0VAL

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	CH3VAL Channel 3 current value	0x0	R	**Channel Value**
2	CH2VAL Channel 2 current value	0x0	R	**Channel Value**
1	CH1VAL Channel 1 current value	0x0	R	**Channel Value**
0	CH0VAL Channel 0 current value	0x0	R	**Channel Value**

14.5.6 PRS_ASYNC_CHx_CTRL - Async Channel Control Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0								0xC					0x0												0x0		
Access					RW								RW					RW												RW		
Name					AUXSEL								FNSEL					SOURCESEL												SIGSEL		

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	AUXSEL	0x0	RW	**Auxiliary LUT Input Select**
	Select Asynchronous PRS channel as input B of LUT function. Async PRS[n] is selected with AUXSEL = n.			
23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	FNSEL	0xC	RW	**Function Select**
	Look up table function select. Signal A is the selected producer input. Signal B is the output of the previous PRS channel.			

Value	Mode	Description
0	LOGICAL_ZERO	Logical 0
1	A_NOR_B	A NOR B
2	NOT_A_AND_B	(!A) AND B
3	NOT_A	!A
4	A_AND_NOT_B	A AND (!B)
5	NOT_B	!B
6	A_XOR_B	A XOR B
7	A_NAND_B	A NAND B
8	A_AND_B	A AND B
9	A_XNOR_B	A XNOR B
10	B	B
11	NOT_A_OR_B	(!A) OR B
12	A	A
13	A_OR_NOT_B	A OR (!B)
14	A_OR_B	A OR B
15	LOGICAL_ONE	Logical 1

15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:8	SOURCESEL	0x0	RW	**Source Select**
	Select input source for asynchronous PRS channel. See Asynchronous Producers table for details.			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	SIGSEL	0x0	RW	**Signal Select**

Bit	Name	Reset	Access	Description
	Select signal input for asynchronous PRS channel. See Asynchronous Producers table for details.			

Value	Mode	Description
0	NONE	

14.5.7 PRS_SYNC_CHx_CTRL - Sync Channel Control Register

Offset	Bit Position																															
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																		0x0												0x0		
Access																		RW												RW		
Name																		SOURCESEL												SIGSEL		

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:8	SOURCESEL	0x0	RW	**Source Select**
	Select input source to sync PRS channel.			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	SIGSEL	0x0	RW	**Signal Select**
	Select signal input to sync PRS channel.			

Value	Mode	Description
0	NONE	

14.5.8 PRS_CONSUMER_EUSART0_CLK - CLK Consumer Register

Offset	Bit Position																															
0x068	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		
Access																													RW		
Name																													PRSSEL		

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CLK async channel select**
	CLK async channel select			

14.5.9 PRS_CONSUMER_EUSART0_RX - RX Consumer Register

Offset	Bit Position																															
0x06C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL RX async channel select	0x0	RW	**RX async channel select**

14.5.10 PRS_CONSUMER_EUSART0_TRIGGER - TRIGGER Consumer Register

Offset	Bit Position																															
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL TRIGGER async channel select	0x0	RW	**TRIGGER async channel select**

14.5.11 PRS_CONSUMER_EUSART1_CLK - CLK Consumer Register

Offset	Bit Position																															
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CLK async channel select**
	CLK async channel select			

14.5.12 PRS_CONSUMER_EUSART1_RX - RX Consumer Register

Offset	Bit Position																															
0x078	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**RX async channel select**
	RX async channel select			

14.5.13 PRS_CONSUMER_EUSART1_TRIGGER - TRIGGER Consumer Register

Offset	Bit Position																															
0x07C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TRIGGER async channel select**
	TRIGGER async channel select			

14.5.14 PRS_CONSUMER_EUSART2_CLK - CLK Consumer Register

Offset	Bit Position																															
0x080	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CLK async channel select**
	CLK async channel select			

14.5.15 PRS_CONSUMER_EUSART2_RX - RX Consumer Register

Offset	Bit Position																															
0x084	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL RX async channel select	0x0	RW	**RX async channel select**

14.5.16 PRS_CONSUMER_EUSART2_TRIGGER - TRIGGER Consumer Register

Offset	Bit Position																															
0x088	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL TRIGGER async channel select	0x0	RW	**TRIGGER async channel select**

14.5.17 PRS_CONSUMER_FRC_RXRAW - RXRAW Consumer Register

Offset	Bit Position																															
0x08C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**RXRAW async channel select**
	RXRAW async channel select			

14.5.18 PRS_CONSUMER_CMU_CALDN - CALDN Consumer Register

Offset	Bit Position																															
0x090	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CALDN async channel select**
	CALDN async channel select			

14.5.19 PRS_CONSUMER_CMU_CALUP - CALUP Consumer Register

Offset	Bit Position																															
0x094	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CALUP async channel select**
	CALUP async channel select			

14.5.20 PRS_CONSUMER_LDMAXBAR0_DMAREQ0 - DMAREQ0 Consumer Register

Offset	Bit Position																															
0x098	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DMAREQ0 async channel select**
	DMAREQ0 async channel select			

14.5.21 PRS_CONSUMER_LDMAXBAR0_DMAREQ1 - DMAREQ1 Consumer Register

Offset	Bit Position																															
0x09C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DMAREQ1 async channel select**
	DMAREQ1 async channel select			

14.5.22 PRS_CONSUMER_LEDDRV0_LEDDRV0ASYNC - LEDDRV0 Consumer Register

Offset	Bit Position																															
0x0A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**LEDDRV0 async channel select**
	LEDDRV0 async channel select			

14.5.23 PRS_CONSUMER_LEDDRV0_LEDDRV1ASYNC - LEDDRV1 Consumer Register

Offset	Bit Position																															
0x0A4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**LEDDRV1 async channel select**
	LEDDRV1 async channel select			

14.5.24 PRS_CONSUMER_LETIMER0_CLEAR - CLEAR Consumer Register

Offset	Bit Position																															
0x0A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CLEAR async channel select**
	CLEAR async channel select			

14.5.25 PRS_CONSUMER_LETIMER0_START - START Consumer Register

Offset	Bit Position																															
0x0AC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**START async channel select**
	START async channel select			

14.5.26 PRS_CONSUMER_LETIMER0_STOP - STOP Consumer Register

Offset	Bit Position																															
0x0B0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**STOP async channel select**
	STOP async channel select			

14.5.27 PRS_CONSUMER_MODEM_DIN - DIN Consumer Register

Offset	Bit Position																															
0x0B4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DIN async channel select**
	DIN async channel select			

14.5.28 PRS_CONSUMER_MODEM_PAEN - PAEN Consumer Register

Offset	Bit Position																															
0x0B8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**PAEN async channel select**
	PAEN async channel select			

14.5.29 PRS_CONSUMER_PCNT0_S0IN - S0IN Consumer Register

Offset	Bit Position																															
0x0BC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**S0IN async channel select**
	S0IN async channel select			

14.5.30 PRS_CONSUMER_PCNT0_S1IN - S1IN Consumer Register

Offset	Bit Position																															
0x0C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**S1IN async channel select**
	S1IN async channel select			

14.5.31 PRS_CONSUMER_PIXELRZ0_INASYNC - IN Consumer Register

Offset	Bit Position																															
0x0C4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL IN async channel select	0x0	RW	**IN async channel select**

14.5.32 PRS_CONSUMER_PIXELRZ1_INASYNC - IN Consumer Register

Offset	Bit Position																															
0x0C8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL IN async channel select	0x0	RW	**IN async channel select**

14.5.33 PRS_CONSUMER_RAC_CLR - CLR Consumer Register

Offset	Bit Position																															
0x0F8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CLR async channel select**
	CLR async channel select			

14.5.34 PRS_CONSUMER_RAC_CTIIN0 - CTI Consumer Register

Offset	Bit Position																															
0x0FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CTI async channel select**
	CTI async channel select			

14.5.35 PRS_CONSUMER_RAC_CTIIN1 - CTI Consumer Register

Offset	Bit Position																															
0x100	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL CTI async channel select	0x0	RW	**CTI async channel select**

14.5.36 PRS_CONSUMER_RAC_CTIIN2 - CTI Consumer Register

Offset	Bit Position																															
0x104	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL CTI async channel select	0x0	RW	**CTI async channel select**

14.5.37 PRS_CONSUMER_RAC_CTIIN3 - CTI Consumer Register

Offset	Bit Position																															
0x108	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CTI async channel select**
	CTI async channel select			

14.5.38 PRS_CONSUMER_RAC_FORCETX - FORCETX Consumer Register

Offset	Bit Position																															
0x10C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**FORCETX async channel select**
	FORCETX async channel select			

14.5.39 PRS_CONSUMER_RAC_RXDIS - RXDIS Consumer Register

Offset	Bit Position																															
0x110	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**RXDIS async channel select**
	RXDIS async channel select			

14.5.40 PRS_CONSUMER_RAC_RXEN - RXEN Consumer Register

Offset	Bit Position																															
0x114	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**RXEN async channel select**
	RXEN async channel select			

14.5.41 PRS_CONSUMER_RAC_TXEN - TXEN Consumer Register

Offset	Bit Position																															
0x118	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TXEN async channel select**
	TXEN async channel select			

14.5.42 PRS_CONSUMER_RFTIMER_PULSE - PULSE Consumer Register

Offset	Bit Position																															
0x11C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**PULSE async channel select**
	PULSE async channel select			

14.5.43 PRS_CONSUMER_ADC0_ASYNCSEQ - ASYNC Consumer Register

Offset	Bit Position																															
0x120	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**ASYNC async channel select**
	ASYNC async channel select			

14.5.44 PRS_CONSUMER_ADC0_SYNCSEQ - SYNC Consumer Register

Offset	Bit Position																															
0x124	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0									
Access																							RW									
Name																							SPRSSEL									

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**SYNC sync channel select**
	SYNC sync channel select			
7:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

14.5.45 PRS_CONSUMER_SEQACC_WAIT0 - WAIT0 Consumer Register

Offset	Bit Position																															
0x128	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**WAIT0 async channel select**
	WAIT0 async channel select			

14.5.46 PRS_CONSUMER_SEQACC_WAIT1 - WAIT1 Consumer Register

Offset	Bit Position																															
0x12C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**WAIT1 async channel select**
	WAIT1 async channel select			

14.5.47 PRS_CONSUMER_SEQACC_WAIT2 - WAIT2 Consumer Register

Offset	Bit Position																															
0x130	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**WAIT2 async channel select**
	WAIT2 async channel select			

14.5.48 PRS_CONSUMER_SEQACC_WAIT3 - WAIT3 Consumer Register

Offset	Bit Position																															
0x134	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**WAIT3 async channel select**
	WAIT3 async channel select			

14.5.49 PRS_CONSUMER_SETAMPER_TAMPERSRC29 - TAMPERSRC29 Consumer Register

Offset	Bit Position																															
0x138	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TAMPERSRC29 async channel select**
	TAMPERSRC29 async channel select			

14.5.50 PRS_CONSUMER_SETAMPER_TAMPERSRC30 - TAMPERSRC30 Consumer Register

Offset	Bit Position																															
0x13C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TAMPERSRC30 async channel select**
	TAMPERSRC30 async channel select			

14.5.51 PRS_CONSUMER_SETAMPER_TAMPERSRC31 - TAMPERSRC31 Consumer Register

Offset	Bit Position																															
0x140	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TAMPERSRC31 async channel select**
	TAMPERSRC31 async channel select			

14.5.52 PRS_CONSUMER_SYSRTC0_IN0 - IN0 Consumer Register

Offset	Bit Position																															
0x144	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**IN0 async channel select**
	IN0 async channel select			

14.5.53 PRS_CONSUMER_SYSRTC0_IN1 - IN1 Consumer Register

Offset	Bit Position																															
0x148	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**IN1 async channel select**
	IN1 async channel select			

14.5.54 PRS_CONSUMER_HFXO0_OSCREQ - OSC Consumer Register

Offset	Bit Position																															
0x14C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**OSC async channel select**
	OSC async channel select			

14.5.55 PRS_CONSUMER_HFXO0_TIMEOUT - TIMEOUT Consumer Register

Offset	Bit Position																															
0x150	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**TIMEOUT async channel select**
	TIMEOUT async channel select			

14.5.56 PRS_CONSUMER_CORE_CTIIN0 - CTI Consumer Register

Offset	Bit Position																															
0x154	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CTI async channel select**
	CTI async channel select			

14.5.57 PRS_CONSUMER_CORE_CTIIN1 - CTI Consumer Register

Offset	Bit Position																															
0x158	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CTI async channel select**
	CTI async channel select			

14.5.58 PRS_CONSUMER_CORE_CTIIN2 - CTI Consumer Register

Offset	Bit Position																															
0x15C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CTI async channel select**
	CTI async channel select			

14.5.59 PRS_CONSUMER_CORE_CTIIN3 - CTI Consumer Register

Offset	Bit Position																															
0x160	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CTI async channel select**
	CTI async channel select			

14.5.60 PRS_CONSUMER_CORE_M33RXEV - M33 Consumer Register

Offset	Bit Position																															
0x164	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**M33 async channel select**
	M33 async channel select			

14.5.61 PRS_CONSUMER_TIMER0_CC0 - CC0 Consumer Register

Offset	Bit Position																															
0x168	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC0 sync channel select**
	CC0 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC0 async channel select**
	CC0 async channel select			

14.5.62 PRS_CONSUMER_TIMER0_CC1 - CC1 Consumer Register

Offset	Bit Position																															
0x16C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC1 sync channel select**
	CC1 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC1 async channel select**
	CC1 async channel select			

14.5.63 PRS_CONSUMER_TIMER0_CC2 - CC2 Consumer Register

Offset	Bit Position																															
0x170	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC2 sync channel select**
	CC2 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC2 async channel select**
	CC2 async channel select			

14.5.64 PRS_CONSUMER_TIMER0_DTI - DTI Consumer Register

Offset	Bit Position																															
0x174	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

14.5.65 PRS_CONSUMER_TIMER0_DTIFS1 - DTI Consumer Register

Offset	Bit Position																															
0x178	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

14.5.66 PRS_CONSUMER_TIMER0_DTIFS2 - DTI Consumer Register

Offset	Bit Position																															
0x17C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

14.5.67 PRS_CONSUMER_TIMER1_CC0 - CC0 Consumer Register

Offset	Bit Position																															
0x18C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC0 sync channel select**
	CC0 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC0 async channel select**
	CC0 async channel select			

14.5.68 PRS_CONSUMER_TIMER1_CC1 - CC1 Consumer Register

Offset	Bit Position																															
0x190	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC1 sync channel select**
	CC1 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC1 async channel select**
	CC1 async channel select			

14.5.69 PRS_CONSUMER_TIMER1_CC2 - CC2 Consumer Register

Offset	Bit Position																															
0x194	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC2 sync channel select**
	CC2 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC2 async channel select**
	CC2 async channel select			

14.5.70 PRS_CONSUMER_TIMER1_DTI - DTI Consumer Register

Offset	Bit Position																															
0x198	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

14.5.71 PRS_CONSUMER_TIMER1_DTIFS1 - DTI Consumer Register

Offset	Bit Position																															
0x19C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL DTI async channel select	0x0	RW	**DTI async channel select**

14.5.72 PRS_CONSUMER_TIMER1_DTIFS2 - DTI Consumer Register

Offset	Bit Position																															
0x1A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL DTI async channel select	0x0	RW	**DTI async channel select**

14.5.73 PRS_CONSUMER_TIMER2_CC0 - CC0 Consumer Register

Offset	Bit Position																															
0x1A4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC0 sync channel select**
	CC0 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC0 async channel select**
	CC0 async channel select			

14.5.74 PRS_CONSUMER_TIMER2_CC1 - CC1 Consumer Register

Offset	Bit Position																															
0x1A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC1 sync channel select**
	CC1 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC1 async channel select**
	CC1 async channel select			

14.5.75 PRS_CONSUMER_TIMER2_CC2 - CC2 Consumer Register

Offset	Bit Position																															
0x1AC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC2 sync channel select**
	CC2 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC2 async channel select**
	CC2 async channel select			

14.5.76 PRS_CONSUMER_TIMER2_CC3 - CC3 Consumer Register

Offset	Bit Position																															
0x1B0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC3 sync channel select**
	CC3 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC3 async channel select**
	CC3 async channel select			

14.5.77 PRS_CONSUMER_TIMER2_CC4 - CC4 Consumer Register

Offset	Bit Position																															
0x1B4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC4 sync channel select**
	CC4 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC4 async channel select**
	CC4 async channel select			

14.5.78 PRS_CONSUMER_TIMER2_CC5 - CC5 Consumer Register

Offset	Bit Position																															
0x1B8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC5 sync channel select**
	CC5 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC5 async channel select**
	CC5 async channel select			

14.5.79 PRS_CONSUMER_TIMER2_CC6 - CC6 Consumer Register

Offset	Bit Position																															
0x1BC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC6 sync channel select**
	CC6 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC6 async channel select**
	CC6 async channel select			

14.5.80 PRS_CONSUMER_TIMER2_DTI - DTI Consumer Register

Offset	Bit Position																															
0x1C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

14.5.81 PRS_CONSUMER_TIMER2_DTIFS1 - DTI Consumer Register

Offset	Bit Position																															
0x1C4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL DTI async channel select	0x0	RW	**DTI async channel select**

14.5.82 PRS_CONSUMER_TIMER2_DTIFS2 - DTI Consumer Register

Offset	Bit Position																															
0x1C8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL DTI async channel select	0x0	RW	**DTI async channel select**

14.5.83 PRS_CONSUMER_TIMER3_CC0 - CC0 Consumer Register

Offset	Bit Position																															
0x1CC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC0 sync channel select**
	CC0 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC0 async channel select**
	CC0 async channel select			

14.5.84 PRS_CONSUMER_TIMER3_CC1 - CC1 Consumer Register

Offset	Bit Position																															
0x1D0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC1 sync channel select**
	CC1 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC1 async channel select**
	CC1 async channel select			

14.5.85 PRS_CONSUMER_TIMER3_CC2 - CC2 Consumer Register

Offset	Bit Position																															
0x1D4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC2 sync channel select**
	CC2 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC2 async channel select**
	CC2 async channel select			

14.5.86 PRS_CONSUMER_TIMER3_CC3 - CC3 Consumer Register

Offset	Bit Position																															
0x1D8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC3 sync channel select**
	CC3 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC3 async channel select**
	CC3 async channel select			

14.5.87 PRS_CONSUMER_TIMER3_CC4 - CC4 Consumer Register

Offset	Bit Position																															
0x1DC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC4 sync channel select**
	CC4 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC4 async channel select**
	CC4 async channel select			

14.5.88 PRS_CONSUMER_TIMER3_CC5 - CC5 Consumer Register

Offset	Bit Position																															
0x1E0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC5 sync channel select**
	CC5 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC5 async channel select**
	CC5 async channel select			

14.5.89 PRS_CONSUMER_TIMER3_CC6 - CC6 Consumer Register

Offset	Bit Position																															
0x1E4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0						0x0			
Access																							RW						RW			
Name																							SPRSSEL						PRSSEL			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	SPRSSEL	0x0	RW	**CC6 sync channel select**
	CC6 sync channel select			
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**CC6 async channel select**
	CC6 async channel select			

14.5.90 PRS_CONSUMER_TIMER3_DTI - DTI Consumer Register

Offset	Bit Position																															
0x1E8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

14.5.91 PRS_CONSUMER_TIMER3_DTIFS1 - DTI Consumer Register

Offset	Bit Position																															
0x1EC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

14.5.92 PRS_CONSUMER_TIMER3_DTIFS2 - DTI Consumer Register

Offset	Bit Position																															
0x1F0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**DTI async channel select**
	DTI async channel select			

14.5.93 PRS_CONSUMER_WDOG0_SRC0 - SRC0 Consumer Register

Offset	Bit Position																															
0x1F4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**SRC0 async channel select**
	SRC0 async channel select			

14.5.94 PRS_CONSUMER_WDOG0_SRC1 - SRC1 Consumer Register

Offset	Bit Position																															
0x1F8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**SRC1 async channel select**
	SRC1 async channel select			

14.5.95 PRS_CONSUMER_WDOG1_SRC0 - SRC0 Consumer Register

Offset	Bit Position																															
0x1FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**SRC0 async channel select**
	SRC0 async channel select			

14.5.96 PRS_CONSUMER_WDOG1_SRC1 - SRC1 Consumer Register

Offset	Bit Position																															
0x200	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													PRSSEL			

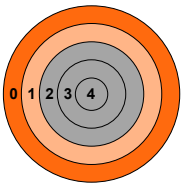
Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	PRSSEL	0x0	RW	**SRC1 async channel select**
	SRC1 async channel select			

14.5.97 PRS_RPURATD4 - Root Access Type Descriptor Register

Offset	Bit Position																															
0x214	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																RATDWDOG1SRC1

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	RATDWDOG1SRC1	0x0	RW	**WDOG1_SRC1 Protection Bit** If LOCKED(1), write access to WDOG1_SRC1 is locked for user master but allowed to root masters. Read allowed for any.

15. GPCRC - General Purpose Cyclic Redundancy Check



Quick Facts

What?

The GPCRC is an error-detecting module commonly used in digital networks and storage systems to detect accidental changes to data.

Why?

The GPCRC module can detect errors in data, giving a higher system reliability and robustness.

How?

Blocks of data entering GPCRC module can have a short checksum, based on the remainder of a polynomial division of their contents; on retrieval the calculation is repeated, and corrective action can be taken against presumed data corruption if the check values do not match.

15.1 Introduction

The GPCRC module implements a Cyclic Redundancy Check (CRC) function. It supports both 32-bit and 16-bit polynomials. The supported 32-bit polynomial is 0x04C11DB7(IEEE 802.3), while the 16-bit polynomial can be programmed to any value, depending on the needs of the application. Common 16-bit polynomials are 0x1021 (CCITT-16), 0x3D65 (IEC16-MBus), and 0x8005 (zigbee, 802.15.4, and USB).

15.2 Features

- Programmable 16-bit polynomial, fixed 32-bit polynomial
- Byte-level bit reversal for the CRC input
- Byte-order reorientation for the CRC input
- Word or half-word bit reversal of the CRC result
- Ability to configure and seed an operation in a single register write
- Single-cycle CRC computation for 32-, 16-, or 8-bit blocks
- DMA operation

15.3 Functional Description

An overview of the GPCRC module is shown in Figure 15.1. GPCRC Overview.

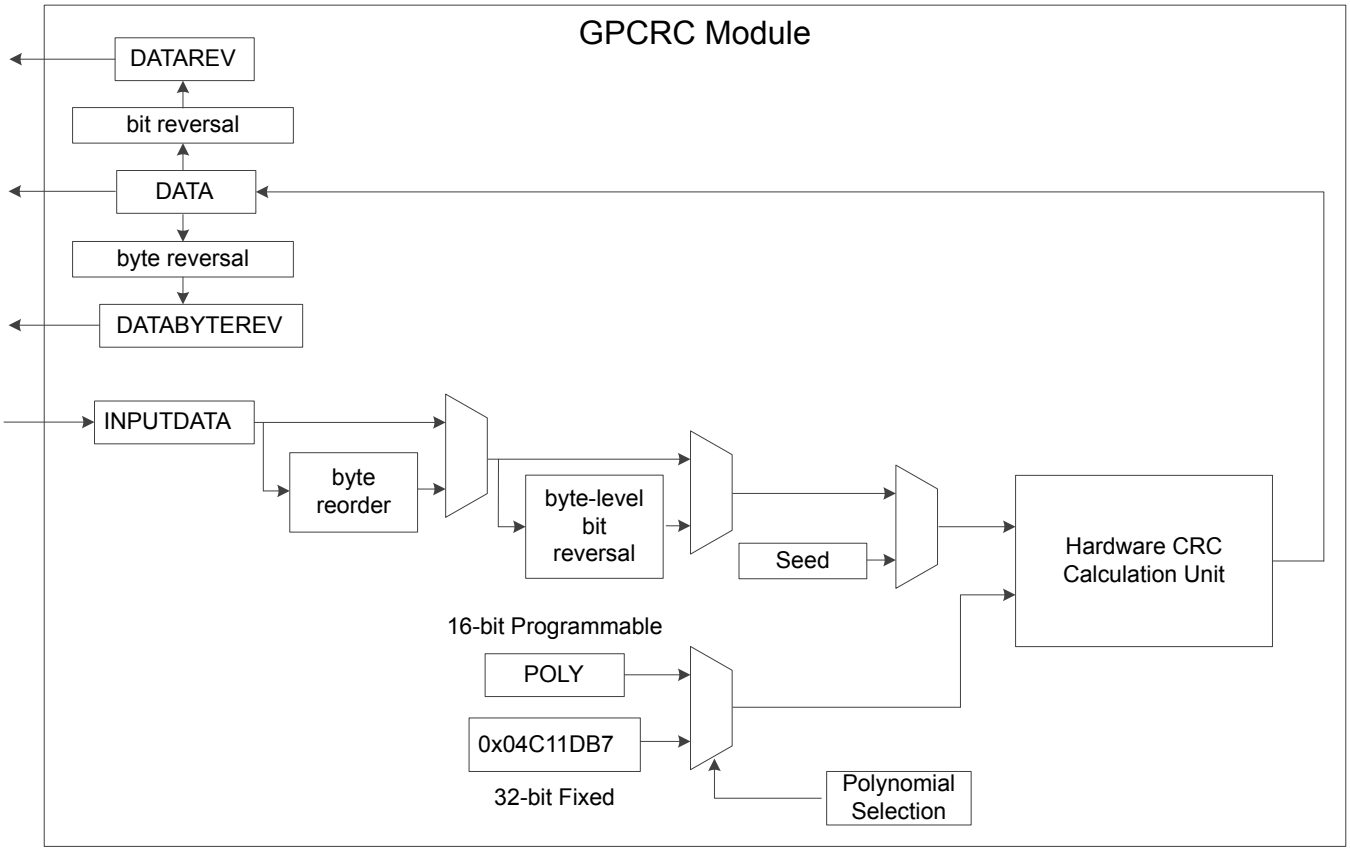


Figure 15.1. GPCRC Overview

15.3.1 Polynomial Specification

POLYSEL in GPCRC_CTRL selects between 32-bit and 16-bit polynomial functions. When a 32-bit polynomial is selected, the fixed IEEE 802.3 polynomial(0x04C11DB7) is used. When a 16-bit polynomial is selected, any valid polynomial can be defined by the user in GPCRC_POLY.

A valid 16-bit CRC polynomial must have an x^16 term and an x^0 term. Theoretically, a 16-bit polynomial has 17 terms total. The convention used is to omit the x^16 term. The polynomial should be written in **reversed** (little endian) bit order. The most significant bit corresponds to the lowest order term. Thus, the most significant bit in GPCRC_POLY represents the x^0 term, and the least significant bit in GPCRC_POLY represents the x^15 term. The highest significant bit of GPCRC_POLY should always set to 1.The polynomial representation for the CRC-16-CCIT polynomial x^16 + x^12 + x^5 + 1, or 0x8408 in reversed order, is shown in Figure 15.2. Polynomial Representation.

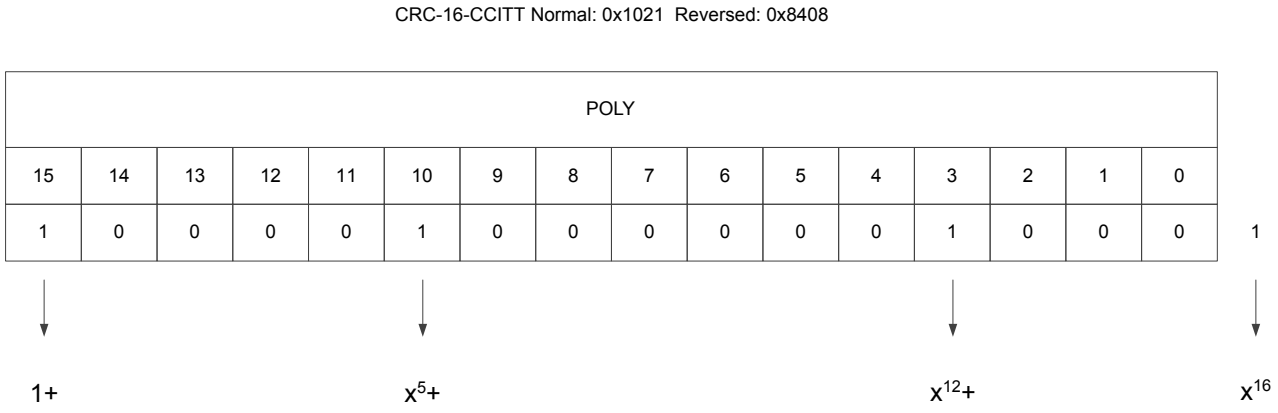


Figure 15.2. Polynomial Representation

15.3.2 Input and Output Specification

The CRC input data can be written to the GPCRC_INPUTDATA, GPCRC_INPUTDATAHWORD or GPCRC_INPUTDATABYTE register via the APB bus based on different data size. If BYTEMODE in GPCRC_CTRL is set, only the least significant byte of the data word will be used for the CRC calculation no matter which input register is written. There are also three output registers for different ordering. Reading from GPCRC_DATA will get the result based on the polynomial in reversed order, while reading from GPCRC_DATAREV will get the result based on the polynomial in normal order. The CRC calculation completes in one clock cycle. Reads from the GPCRC_DATA, GPCRC_DATAREV or GPCRC_DATABYTEREV registers and writes to the GPCRC_CMD register are halted while the calculation is in progress.

.

15.3.3 Initialization

The CRC can be pre-loaded or re-initialized by first writing a 32-bit programmable init value to INIT in GPCRC_INIT and then setting INIT in GPCRC_CMD. It can also be re-initialized automatically when read from DATA, DATAREV or DATABYTEREV provided that AUTOINIT in GPCRC_CTRL is set, the CRC would be re-initialized with the stored init value.

15.3.4 DMA Usage

A DMA channel may be used to transfer data into the CRC engine. All bytes and half-word writes must be word-aligned. The recommended DMA usage model is to use the DMA to transfer all available words of data and use software writes to capture any remaining bytes.

15.3.5 Byte-Level Bit Reversal and Byte Reordering

The byte-level bit reversal and byte reordering operations occur before the data is used in the CRC calculation. Byte reordering can occur on words or half words. The hardware ignores the BYTEREVERSE field with any byte writes or operations with byte mode enabled (BYTEMODE = 1), but the bit reversal settings (BITREVERSE) are still applied to the byte. 32-bit little endian MSB-first data can be treated like 32-bit little endian LSB-first data, as shown in Figure 15.3. Data Ordering Example - 32-bit MSB -first to LSB-first. In this example, 32-bit data is written to GPCRC_INPUTDATA, BYTEREVERSE is set for byte ordering, and BITREVERSE is set for byte-level bit reversal.

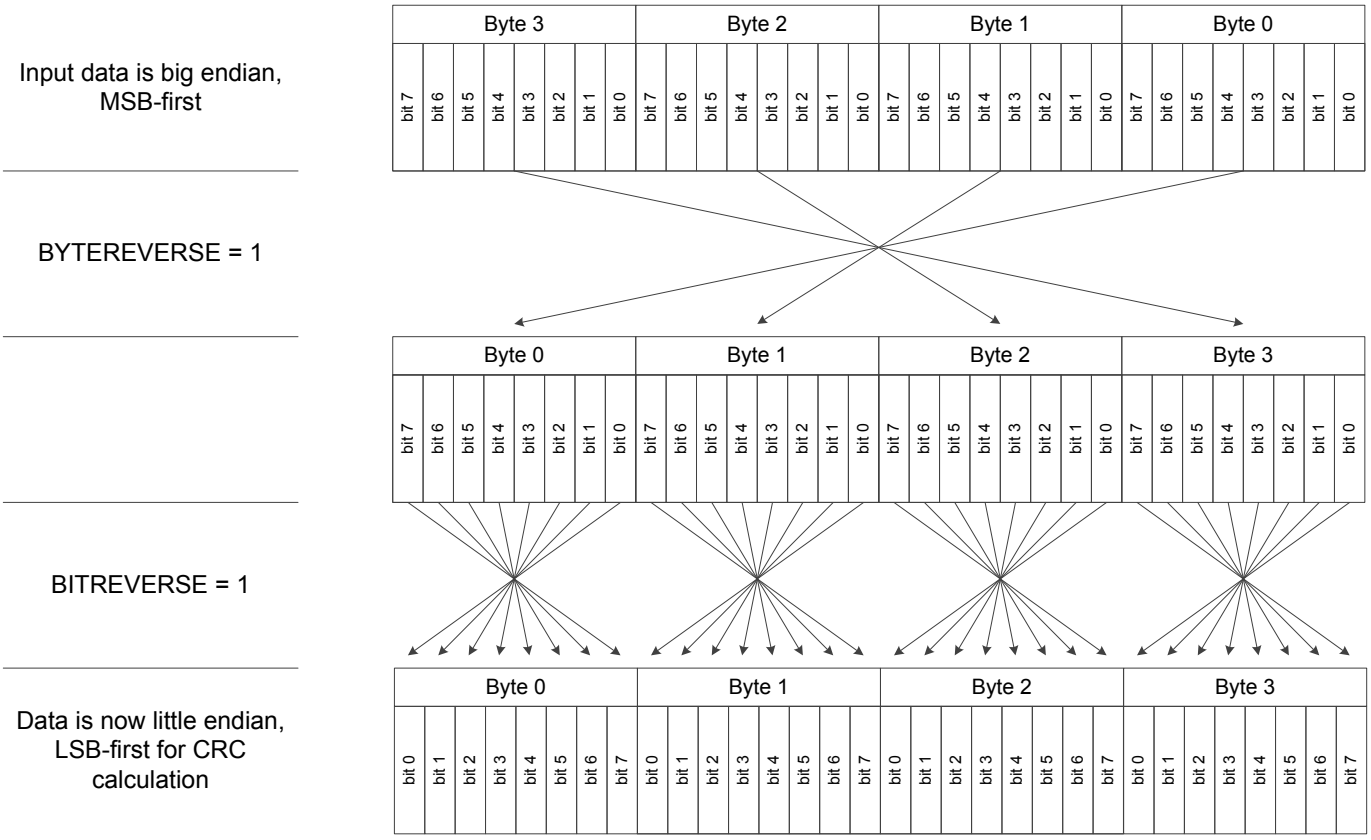


Figure 15.3. Data Ordering Example - 32-bit MSB -first to LSB-first

When handling 16-bit data, the byte reordering function only swap the two lowest bytes and clear the two highest bytes, as shown in Figure 15.4. Data Ordering Example - 16-bit MSB -first to LSB-first. In this example, 16-bit data is written to GPCRC_INPUTDATAH-WORD, BYTEREVERSE is set for byte ordering, and BITREVERSE is set for byte-level bit reversal.

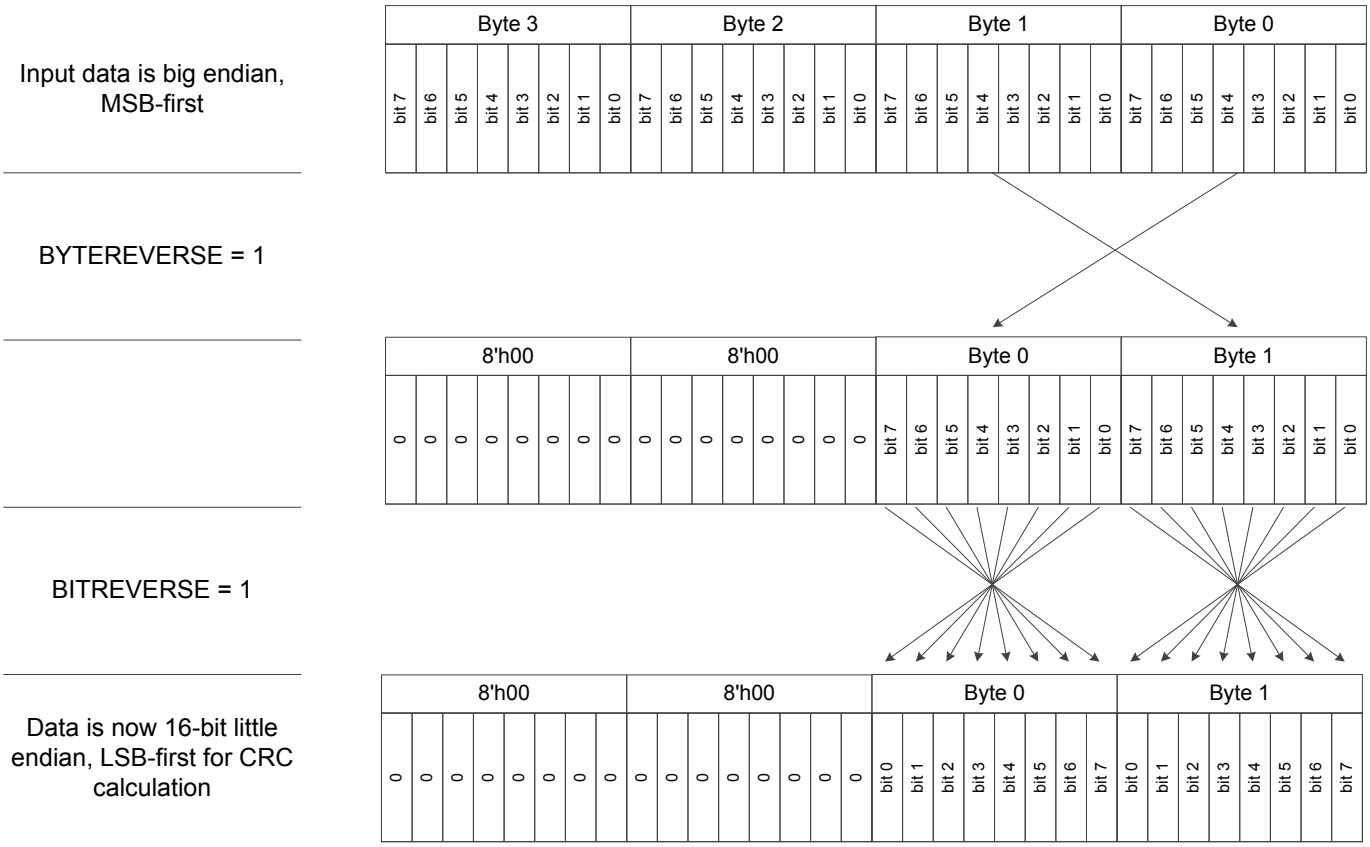


Figure 15.4. Data Ordering Example - 16-bit MSB -first to LSB-first

Assuming a word input byte order of B3 B2 B1 B0, the values used in the CRC calculation for the various settings of the byte-level bit reversal and byte reordering are shown in Table 15.1. Byte-Level Bit Reversal and Byte Reordering Results (B3 B2 B1 B0 Input Order).

Table 15.1. Byte-Level Bit Reversal and Byte Reordering Results (B3 B2 B1 B0 Input Order)

Input Width(bits)	BYTEREVERSE Setting	BITREVERSE Setting	Input to CRC Calculation
32	0	0	B3 B2 B1 B0
32	1	1	'B0 'B1 'B2 'B3
32	1	0	B0 B1 B2 B3
32	0	1	'B3 'B2 'B1 'B0
16	0	0	XX XX B1 B0
16	1	1	XX XX 'B0 'B1
16	1	0	XX XX B0 B1
16	0	1	XX XX 'B1 'B0
8	-	0	XX XX XX XX B0
8	-	1	XX XX XX XX 'B0

Notes:

1. X indicates a "don't care".
2. Bn is the byte field within the word.
3. 'Bn is the bit-reversed byte field within the word.

15.4 GPCRC Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	GPCRC_IPVERSION	R	IPVERSION
0x004	GPCRC_EN	RW	CRC Enable
0x008	GPCRC_CTRL	RW	Control Register
0x00C	GPCRC_CMD	W	Command Register
0x010	GPCRC_INIT	RWH	CRC Init Value
0x014	GPCRC_POLY	RW	CRC Polynomial Value
0x018	GPCRC_INPUTDATA	W	Input 32-Bit Data Register
0x01C	GPCRC_INPUTDATAHWORD	W	Input 16-Bit Data Register
0x020	GPCRC_INPUTDATABYTE	W	Input 8-Bit Data Register
0x024	GPCRC_DATA	RH(r)	CRC Data Register
0x028	GPCRC_DATAREV	RH(r)	CRC Data Reverse Register
0x02C	GPCRC_DATABYTEREV	RH(r)	CRC Data Byte Reverse Register
0x1000	GPCRC_IPVERSION_SET	R	IPVERSION
0x1004	GPCRC_EN_SET	RW	CRC Enable
0x1008	GPCRC_CTRL_SET	RW	Control Register
0x100C	GPCRC_CMD_SET	W	Command Register
0x1010	GPCRC_INIT_SET	RWH	CRC Init Value
0x1014	GPCRC_POLY_SET	RW	CRC Polynomial Value
0x1018	GPCRC_INPUTDATA_SET	W	Input 32-Bit Data Register
0x101C	GPCRC_INPUTDATAH-WORD_SET	W	Input 16-Bit Data Register
0x1020	GPCRC_INPUTDATABYTE_SET	W	Input 8-Bit Data Register
0x1024	GPCRC_DATA_SET	RH(r)	CRC Data Register
0x1028	GPCRC_DATAREV_SET	RH(r)	CRC Data Reverse Register
0x102C	GPCRC_DATABYTEREV_SET	RH(r)	CRC Data Byte Reverse Register
0x2000	GPCRC_IPVERSION_CLR	R	IPVERSION
0x2004	GPCRC_EN_CLR	RW	CRC Enable
0x2008	GPCRC_CTRL_CLR	RW	Control Register
0x200C	GPCRC_CMD_CLR	W	Command Register
0x2010	GPCRC_INIT_CLR	RWH	CRC Init Value
0x2014	GPCRC_POLY_CLR	RW	CRC Polynomial Value
0x2018	GPCRC_INPUTDATA_CLR	W	Input 32-Bit Data Register
0x201C	GPCRC_INPUTDATAH-WORD_CLR	W	Input 16-Bit Data Register
0x2020	GPCRC_INPUTDATABYTE_CLR	W	Input 8-Bit Data Register
0x2024	GPCRC_DATA_CLR	RH(r)	CRC Data Register
0x2028	GPCRC_DATAREV_CLR	RH(r)	CRC Data Reverse Register
0x202C	GPCRC_DATABYTEREV_CLR	RH(r)	CRC Data Byte Reverse Register
0x3000	GPCRC_IPVERSION_TGL	R	IPVERSION
0x3004	GPCRC_EN_TGL	RW	CRC Enable

Offset	Name	Type	Description
0x3008	GPCRC_CTRL_TGL	RW	Control Register
0x300C	GPCRC_CMD_TGL	W	Command Register
0x3010	GPCRC_INIT_TGL	RWH	CRC Init Value
0x3014	GPCRC_POLY_TGL	RW	CRC Polynomial Value
0x3018	GPCRC_INPUTDATA_TGL	W	Input 32-Bit Data Register
0x301C	GPCRC_INPUTDATAH-WORD_TGL	W	Input 16-Bit Data Register
0x3020	GPCRC_INPUTDATABYTE_TGL	W	Input 8-Bit Data Register
0x3024	GPCRC_DATA_TGL	RH(r)	CRC Data Register
0x3028	GPCRC_DATAREV_TGL	RH(r)	CRC Data Reverse Register
0x302C	GPCRC_DATABYTEREV_TGL	RH(r)	CRC Data Byte Reverse Register

15.5 GPCRC Register Description

15.5.1 GPCRC_IPVERSION - IPVERSION

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x0	R	**IPVERSION**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

15.5.2 GPCRC_EN - CRC Enable

Offset	Bit Position																															
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																EN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	EN	0x0	RW	**CRC Enable**
	The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.			

15.5.3 GPCRC_CTRL - Control Register

Offset	Bit Position																														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																			0x0			0x0	0x0	0x0				0x0				
Access																			RW			RW	RW	RW				RW				
Name																			AUTOINIT			BYTEREVERSE	BITREVERSE	BYTEMODE				POLYSEL				

Bit	Name	Reset	Access	Description
31:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	AUTOINIT	0x0	RW	**Auto Init Enable**
	Enables auto init by re-seeding the CRC result based on the value in INIT after reading of DATA, DATAREV or DATABYTEREV.			
12:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	BYTEREVERSE	0x0	RW	**Byte Reverse Mode**
	Allows byte level reverse of bytes B3, B2, B1, B0 within the 32-bit data word			

Value	Mode	Description
0	NORMAL	No reverse: B3, B2, B1, B0
1	REVERSED	Reverse byte order. For 32-bit: B0, B1, B2, B3; For 16-bit: 0, 0, B0, B1

Bit	Name	Reset	Access	Description
9	BITREVERSE	0x0	RW	**Byte-level Bit Reverse Enable**
	Reverses bits within each byte of the 32-bit data word			

Value	Mode	Description
0	NORMAL	No reverse
1	REVERSED	Reverse bit order in each byte

Bit	Name	Reset	Access	Description
8	BYTEMODE	0x0	RW	**Byte Mode Enable**
	Treats all writes as bytes. Only the least significant byte of the data-word will be used for CRC calculation for all writes			
7:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	POLYSEL	0x0	RW	**Polynomial Select**
	Selects 16-bit CRC programmable polynomial or 32-bit CRC fixed polynomial			

Value	Mode	Description
0	CRC32	CRC-32 (0x04C11DB7) polynomial selected
1	CRC16	16-bit CRC programmable polynomial selected

Bit	Name	Reset	Access	Description
3:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

15.5.4 GPCRC_CMD - Command Register

Offset	Bit Position																														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																W
Name																																INIT

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	INIT	0x0	W	**Initialization Enable**
	Writing 1 to this bit initialize the CRC by writing the INIT value in CRC_INIT to CRC_DATA.			

15.5.5 GPCRC_INIT - CRC Init Value

Offset	Bit Position																														
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	INIT																															

Bit	Name	Reset	Access	Description
31:0	INIT	0x0	RW	**CRC Initialization Value**
	This value is loaded into CRC_DATA upon issuing the INIT command in CRC_CMD			

15.5.6 GPCRC_POLY - CRC Polynomial Value

Offset	Bit Position																														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	RW															
Name																	POLY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	POLY	0x0	RW	**CRC Polynomial Value**
	This value defines 16-bit POLY, which is used as the polynomial during the 16-bit CRC calculation. The polynomial is defined in reversed representation, meaning that the lowest degree term is in the highest bit position of POLY. Additionally, the highest degree term in the polynomial is implicit. Further examples of the CRC configuration can be found in the documentation.			

15.5.7 GPCRC_INPUTDATA - Input 32-Bit Data Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	W																															
Name	INPUTDATA																															

Bit	Name	Reset	Access	Description
31:0	INPUTDATA	0x0	W	**Input Data for 32-bit**
	CRC Input 32-bit Data can be written to this register. Each time this register is written, the CRC value is updated.			

15.5.8 GPCRC_INPUTDATAHWORD - Input 16-Bit Data Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	W															
Name																	INPUTDATAHWORD															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	INPUTDATAHWORD	0x0	W	**Input Data for 16-bit**
	CRC Input 16-bit Data can be written to this register. Each time this register is written, the CRC value is updated.			

15.5.9 GPCRC_INPUTDATABYTE - Input 8-Bit Data Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									INPUTDATABYTE							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	INPUTDATABYTE	0x0	W	**Input Data for 8-bit**
	CRC Input 8-bit Data can be written to this register. Each time this register is written, the CRC value is updated.			

15.5.10 GPCRC_DATA - CRC Data Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R(r)																															
Name	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	R(r)	**CRC Data Register**
	CRC Data Register, read only. The CRC data register may still be indirectly written from software, by writing the INIT register and then issue an INITIALIZE command.			

15.5.11 GPCRC_DATAREV - CRC Data Reverse Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R(r)																															
Name	DATAREV																															

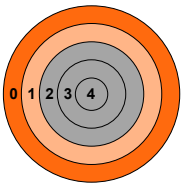
Bit	Name	Reset	Access	Description
31:0	DATAREV	0x0	R(r)	**Data Reverse Value**
	Bit reversed version of CRC Data register. When a 32-bit CRC polynomial is selected, the reversal occurs on the entire 32-bit word. When a 16-bit CRC polynomial is selected, the bits [15:0] are reversed.			

15.5.12 GPCRC_DATABYTEREV - CRC Data Byte Reverse Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R(r)																															
Name	DATABYTEREV																															

Bit	Name	Reset	Access	Description
31:0	DATABYTEREV	0x0	R(r)	**Data Byte Reverse Value**
	Byte reversed version of CRC Data register. When a 32-bit CRC polynomial is selected, the bytes are swizzled to {B0, B1, B2, B3}. When a 16-bit CRC polynomial is selected, the bytes are swizzled to {0, 0, B0, B1}.			

16. SYSRTC - System RTC



Quick Facts

What?

The SYSRTC is a 32-bit Real Time Clock ensuring timekeeping in low energy modes.

Why?

Timekeeping over long time periods while using as little power as possible is required in many low power applications.

How?

The SYSRTC operates a central counter off a low-frequency oscillator source. It has configurable compare and capture channels which can be used to trigger wake-up, generate PRS signalling, or capture system events. The SYSRTC allows the system to stay in low-energy modes for long periods of time and maintain reliable timekeeping.

16.1 Introduction

The SYSRTC (System Real Time Counter), is a 32-bit counter kept running down to energy mode EM1. It can be used as a sleep timer / wakeup source as well as a timekeeping counter during low energy modes. Multiple groups of capture / compare registers are available to different cores in the system, allowing the peripheral and time base to be shared across cores and save energy. Only group 0 registers are directly accessible to the main processor core, but compare / capture signals and interrupts from group 1 are also available to the main processor.

Capture compare channels can be used to trigger interrupts, generate PRS signals, capture PRS events, and to wake the device up from EM0 and EM1.

Note: Critical portions of the EFR32/EFM32 software stack related to system timing and power management make use of SYSRTC in such a way that it is effectively unavailable to user software. Please refer to relevant software documentation for additional information.

16.2 Features

- 32-bit counter
- Debug mode
- 32.768 kHz LFXO or LFRCO / 1 kHz ULFRCO
- Low energy wake-up source
- Millisecond timer
- Separate groups of capture / compare registers and signals
  - MCU core and Radio core each control one group
  - 2 compare channels per group
  - 1 capture channel per group
  - Separate interrupt vectors for MCU and Radio events
  - Capture / compare available in PRS

16.3 Functional Description

16.3.1 Interrupts and Wake Events

Each group has a dedicated, independent interrupt line, and has the ability to wake the system. The Group 0 interrupt line (SYSRTC_APP) is used and fully controlled by user code running on the main processor. The Group 1 interrupt line (SYSRTC_SEQ) is also available to the main processor, but Group 1 registers and configuration are not directly available to the user application.

16.3.2 Counter

A single counter value (CNT) is used across all groups. The SYSRTC module is enabled by setting its EN bit field. The counter value is asserted to 0x00000000 on reset. The counter can be started/stopped by writing to the START/STOP bit fields in the CMD register. The RUNNING field in the STATUS register indicates that the counter is running when it is set.

Once started, the counter increments by 1 on each cycle (typically 32.768 kHz). The counter value can be programmed directly by writing to the CNT register. Once the programmed value is applied and the counter is running, the counter will increment on every clock starting from the newly programmed value.

When the counter reaches its maximum value of 0xFFFFFFFF, an overflow event is generated, followed by a counter wrap-around to its reset value (from which counting continues) and the OVF interrupt flag (OVFIF) on the next cycle. The overflow event is common for all the groups, i.e. OVFIF flags in all groups get set.

The normal operation of SYSRTC is to configure it, enable it, start it, and then leave it running. This should be done by a single core so that other cores only access the registers for their designated group as needed. If SYSRTC needs to be disabled, it is recommended to stop it first using the STOP command.

16.3.3 Compare Events

A compare event for channel "n" of group "x" is generated whenever the counter is RUNNING, the CMPnEN bit is set / enabled in the GRPx_CTRL register, and the CNT value is equal to the GRPx_CMPnVALUE register setting. This event is followed by GRPx_IF.CMPn being set on the next counter clock cycle.

Compare events can be routed as PRS producers on the GRPxOUTn signals. There are several options for the match action, selected by CMPnCMOA in GRPx_CTRL. Note that when using the PULSE option, the PRS output should already be cleared for the pulse to get set and the PULSE option should remain configured until the pulse is cleared (otherwise if the PULSE option is reprogrammed to the SET option, the "pulse" remains set). A possible use case when using the CMPIF option is to signal early events prior to the following wake-up. After wakeup, the compare flag should be processed and cleared. To avoid a race condition on the PRS output, the compare flag should be cleared away from the next possible compare event.

Note that when setting the compare value to the current counter value, a compare event may not get generated until the counter overflows and reaches the current value again. To generate a compare event quickly, it is recommended to program the compare value to the current counter value + 1. Compare events are group-specific.

16.3.4 Capture Events

SYSRTC groups support counter value capture triggered by PRS consumer signals. For group "n" the SYSRTC0 "INn" PRS consumer is used to trigger captures. Capture can be triggered on RISING, FALLING, or BOTH edges, according to the setting programmed in CAP0EDGE of the GRPx_CTRL register. A capture event for group "n" is generated whenever the counter is RUNNING, the CAP0EN bit is set / enabled in the GRPx_CTRL register, and the desired event occurs on the PRS output.

A capture event is followed by GRPx_IF.CAP0 being set after up to 3 counter clock cycles. At the same time the flag is set, the GRPx_CAP0VALUE register captures the current counter value. Note that PRS input edges should not occur more frequently than once every three counter cycles. If the counter is being started/stopped or GRPx_CTRL.CAP0EN / GRPx_CTRL.CAP0EDGE is changed close to the PRS input edge, a race condition may occur. Capture events are group-specific.

16.3.5 SYSRTC Behavior on SWRST/Disablement/STOP

On SWRST / Disablement, the counter is reset to 0x00000000, PRS outputs are cleared, compare/capture events are disabled, compare/capture flags are reset, and the CAP0VALUE register is reset.

When the counter is stopped using the STOP command in the CMD register, all other settings remain unchanged, except that the RUNNING status will return to 0, which blocks any compare/capture events until the counter is started again.

16.3.6 Debug Functionality

By default, the counter value is frozen when the main processor is halted during debugging. The RUNNING status bit is not affected by debug halt, and will continue to indicate that the counter is active. If DEBUGRUN in the CFG register is set, the counter will not halt when the main processor is halted, and SYSRTC will continue to count clocks.

Note that the main processor runs on a much higher frequency than the counter and that the halt condition needs to last long enough (more than 3 counter cycles) for the counter to reach the frozen state.

16.4 SYSRTC Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	SYSRTC_IPVERSION	R	IP VERSION
0x004	SYSRTC_EN	RW ENABLE	Module Enable Register
0x008	SYSRTC_SWRST	RW SWRST	Software Reset Register
0x00C	SYSRTC_CFG	RW CONFIG	Configuration Register
0x010	SYSRTC_CMD	W LFSYNC	Command Register
0x014	SYSRTC_STATUS	RH	Status Register
0x018	SYSRTC_CNT	RWH LFSYNC	Counter Value Register
0x01C	SYSRTC_SYNCBUSY	RH	Synchronization Busy Register
0x020	SYSRTC_LOCK	W	Configuration Lock Register
0x024	SYSRTC_MSCNT	RH	Millisecond Counter Register
0x028	SYSRTC_MSCMPVAL	RWH	Millisecond Compare Register
0x02C	SYSRTC_MSCMPBUF	RW	Millisecond Compare Buffer Register
0x030	SYSRTC_IF	RWH INTFLAG	Interrupt Flag Register
0x034	SYSRTC_IEN	RW	Interrupt Enable Register
0x050	SYSRTC_GRP0_IF	RWH INTFLAG	Group Interrupt Flags
0x054	SYSRTC_GRP0_IEN	RW	Group Interrupt Enables
0x058	SYSRTC_GRP0_CTRL	RW	Group Control Register
0x05C	SYSRTC_GRP0_CMP0VALUE	RW	Compare 0 Value Register
0x060	SYSRTC_GRP0_CMP1VALUE	RW	Compare 1 Value Register
0x064	SYSRTC_GRP0_CAP0VALUE	RH	Capture 0 Value Register
0x068	SYSRTC_GRP0_SYNCBUSY	RH	Synchronization Busy Register
0x06C	SYSRTC_GRP0_PRETRIG	RWH LFSYNC	Pretrigger Register
0x1000	SYSRTC_IPVERSION_SET	R	IP VERSION
0x1004	SYSRTC_EN_SET	RW ENABLE	Module Enable Register
0x1008	SYSRTC_SWRST_SET	RW SWRST	Software Reset Register
0x100C	SYSRTC_CFG_SET	RW CONFIG	Configuration Register
0x1010	SYSRTC_CMD_SET	W LFSYNC	Command Register
0x1014	SYSRTC_STATUS_SET	RH	Status Register
0x1018	SYSRTC_CNT_SET	RWH LFSYNC	Counter Value Register
0x101C	SYSRTC_SYNCBUSY_SET	RH	Synchronization Busy Register
0x1020	SYSRTC_LOCK_SET	W	Configuration Lock Register
0x1024	SYSRTC_MSCNT_SET	RH	Millisecond Counter Register
0x1028	SYSRTC_MSCMPVAL_SET	RWH	Millisecond Compare Register
0x102C	SYSRTC_MSCMPBUF_SET	RW	Millisecond Compare Buffer Register
0x1030	SYSRTC_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1034	SYSRTC_IEN_SET	RW	Interrupt Enable Register
0x1050	SYSRTC_GRP0_IF_SET	RWH INTFLAG	Group Interrupt Flags
0x1054	SYSRTC_GRP0_IEN_SET	RW	Group Interrupt Enables
0x1058	SYSRTC_GRP0_CTRL_SET	RW	Group Control Register

Offset	Name	Type	Description
0x105C	SYSRTC_GRP0_CMP0VAL-UE_SET	RW	Compare 0 Value Register
0x1060	SYSRTC_GRP0_CMP1VAL-UE_SET	RW	Compare 1 Value Register
0x1064	SYSRTC_GRP0_CAP0VAL-UE_SET	RH	Capture 0 Value Register
0x1068	SYSRTC_GRP0_SYN-CBUSY_SET	RH	Synchronization Busy Register
0x106C	SYSRTC_GRP0_PRETRIG_SET	RWH LFSYNC	Pretrigger Register
0x2000	SYSRTC_IPVERSION_CLR	R	IP VERSION
0x2004	SYSRTC_EN_CLR	RW ENABLE	Module Enable Register
0x2008	SYSRTC_SWRST_CLR	RW SWRST	Software Reset Register
0x200C	SYSRTC_CFG_CLR	RW CONFIG	Configuration Register
0x2010	SYSRTC_CMD_CLR	W LFSYNC	Command Register
0x2014	SYSRTC_STATUS_CLR	RH	Status Register
0x2018	SYSRTC_CNT_CLR	RWH LFSYNC	Counter Value Register
0x201C	SYSRTC_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x2020	SYSRTC_LOCK_CLR	W	Configuration Lock Register
0x2024	SYSRTC_MSCNT_CLR	RH	Millisecond Counter Register
0x2028	SYSRTC_MSCMPVAL_CLR	RWH	Millisecond Compare Register
0x202C	SYSRTC_MSCMPBUF_CLR	RW	Millisecond Compare Buffer Register
0x2030	SYSRTC_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2034	SYSRTC_IEN_CLR	RW	Interrupt Enable Register
0x2050	SYSRTC_GRP0_IF_CLR	RWH INTFLAG	Group Interrupt Flags
0x2054	SYSRTC_GRP0_IEN_CLR	RW	Group Interrupt Enables
0x2058	SYSRTC_GRP0_CTRL_CLR	RW	Group Control Register
0x205C	SYSRTC_GRP0_CMP0VAL-UE_CLR	RW	Compare 0 Value Register
0x2060	SYSRTC_GRP0_CMP1VAL-UE_CLR	RW	Compare 1 Value Register
0x2064	SYSRTC_GRP0_CAP0VAL-UE_CLR	RH	Capture 0 Value Register
0x2068	SYSRTC_GRP0_SYN-CBUSY_CLR	RH	Synchronization Busy Register
0x206C	SYSRTC_GRP0_PRETRIG_CLR	RWH LFSYNC	Pretrigger Register
0x3000	SYSRTC_IPVERSION_TGL	R	IP VERSION
0x3004	SYSRTC_EN_TGL	RW ENABLE	Module Enable Register
0x3008	SYSRTC_SWRST_TGL	RW SWRST	Software Reset Register
0x300C	SYSRTC_CFG_TGL	RW CONFIG	Configuration Register
0x3010	SYSRTC_CMD_TGL	W LFSYNC	Command Register
0x3014	SYSRTC_STATUS_TGL	RH	Status Register
0x3018	SYSRTC_CNT_TGL	RWH LFSYNC	Counter Value Register
0x301C	SYSRTC_SYNCBUSY_TGL	RH	Synchronization Busy Register
0x3020	SYSRTC_LOCK_TGL	W	Configuration Lock Register

Offset	Name	Type	Description
0x3024	SYSRTC_MSCNT_TGL	RH	Millisecond Counter Register
0x3028	SYSRTC_MSCMPVAL_TGL	RWH	Millisecond Compare Register
0x302C	SYSRTC_MSCMPBUF_TGL	RW	Millisecond Compare Buffer Register
0x3030	SYSRTC_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3034	SYSRTC_IEN_TGL	RW	Interrupt Enable Register
0x3050	SYSRTC_GRP0_IF_TGL	RWH INTFLAG	Group Interrupt Flags
0x3054	SYSRTC_GRP0_IEN_TGL	RW	Group Interrupt Enables
0x3058	SYSRTC_GRP0_CTRL_TGL	RW	Group Control Register
0x305C	SYSRTC_GRP0_CMP0VAL-UE_TGL	RW	Compare 0 Value Register
0x3060	SYSRTC_GRP0_CMP1VAL-UE_TGL	RW	Compare 1 Value Register
0x3064	SYSRTC_GRP0_CAP0VAL-UE_TGL	RH	Capture 0 Value Register
0x3068	SYSRTC_GRP0_SYN-CBUSY_TGL	RH	Synchronization Busy Register
0x306C	SYSRTC_GRP0_PRETRIG_TGL	RWH LFSYNC	Pretrigger Register

16.5 SYSRTC Register Description

16.5.1 SYSRTC_IPVERSION - IP VERSION

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x2																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x2	R	**IP VERSION**
	Gives access to the IP VERSION of SYSRTC			

16.5.2 SYSRTC_EN - Module Enable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	Set when EN cleared and cleared when the peripheal core reset is finished			
0	EN	0x0	RW	**SYSRTC Enable**
	Enable the SYSRTC by requesting Clk from CMU			

16.5.3 SYSRTC_SWRST - Software Reset Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Software reset busy status**
	When SWRST command is issued, resetting logic sets this status immediately and it is later cleared when the reset process is finished			
0	SWRST	0x0	W	**Software reset command**
	A software reset command field resets the module back to the initial condition, similar to the power-on reset condition			

16.5.4 SYSRTC_CFG - Configuration Register

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																DEBUGRUN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	DEBUGRUN	0x0	RW	**Debug Mode Run Enable**

Set this bit to keep the SYSRTC running during a debug halt.

Value	Mode	Description
0	DISABLE	SYSRTC is frozen in debug mode
1	ENABLE	SYSRTC is running in debug mode

16.5.5 SYSRTC_CMD - Command Register

Offset	Bit Position																															
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													W(nB)	W(nB)	W(nB)	W(nB)
Name																													MSSTOP	MSSTART	STOP	START

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	MSSTOP	0x0	W(nB)	**stop ms counter**
	Write a 1 to stop the ms counter			
2	MSSTART	0x0	W(nB)	**Start ms counter**
	Write a 1 to start the ms counter			
1	STOP	0x0	W(nB)	**Stop SYSRTC**
	Write a 1 to stop the SYSRTC			
0	START	0x0	W(nB)	**Start SYSRTC**
	Write a 1 to start the SYSRTC			

16.5.6 SYSRTC_STATUS - Status Register

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0		0x0	0x0
Access																													R		R	R
Name																													MSRUNNING		LOCKSTATUS	RUNNING

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	MSRUNNING	0x0	R	**MS-Counter running status**
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	LOCKSTATUS	0x0	R	**Lock Status**
	Indicates the current status of SYSRTC Lock			

Value	Mode	Description
0	UNLOCKED	SYSRTC registers are unlocked
1	LOCKED	SYSRTC registers are locked

Bit	Name	Reset	Access	Description
0	RUNNING	0x0	R	**SYSRTC running status**
	Indicates the current status of SYSRTC running			

16.5.7 SYSRTC_CNT - Counter Value Register

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	CNT																															

Bit	Name	Reset	Access	Description
31:0	CNT	0x0	RW	**Counter Value**
	Gives access to the counter value of the SYSRTC.			

16.5.8 SYSRTC_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														R	R	R
Name																														CNT	STOP	START

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	CNT	0x0	R	**Sync busy for CNT bitfield**
	Last writing of CNT is synchronizing to LF clock			
1	STOP	0x0	R	**Sync busy for STOP bitfield**
	Last writing of STOP is synchronizing to LF clock			
0	START	0x0	R	**Sync busy for START bitfield**
	Last writing of START is synchronizing to LF clock			

16.5.9 SYSRTC_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x0	W	**Configuration Lock Key**
	Write any other value than the unlock code to lock EN, SWRST, CFG, CMD, CNT registers from editing. Write the unlock code to unlock.			

Value	Mode	Description
18294	UNLOCK	Write to unlock SYSRTC lockable registers

16.5.10 SYSRTC_MSCNT - Millisecond Counter Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	MSCNT																															

Bit	Name	Reset	Access	Description
31:0	MSCNT	0x0	R	**MS-Counter**
	Gives access to the mscounter value of the SYSRTC.			

16.5.11 SYSRTC_MSCMPVAL - Millisecond Compare Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	MSCMPVAL																															

Bit	Name	Reset	Access	Description
31:0	MSCMPVAL	0x0	RW	**Compare val for MSCounter**
	Write to this compare reg for compare trigger wrt MS Counter			

16.5.12 SYSRTC_MSCMPBUF - Millisecond Compare Buffer Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	MSCMPBUF																															

Bit	Name	Reset	Access	Description
31:0	MSCMPBUF	0x0	RW	**Buffered Compare for MSCounter**
	Updated Compare value with buffered comapre on compare trigger wrt to MS Counter			

16.5.13 SYSRTC_IF - Interrupt Flag Register

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															MSCMP	MSOVF

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	MSCMP	0x0	RW	**MS Counter Compare**
	IRQ on MS-Counter compare event			
0	MSOVF	0x0	RW	**MS Counter Overflow**
	IRQ when MS-Counter overflows			

16.5.14 SYSRTC_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															MSCMP	MSOVF

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	MSCMP	0x0	RW	**MS Counter Compare**
	IRQ on MS-Counter compare event			
0	MSOVF	0x0	RW	**MS Counter Overflow**
	IRQ when MS-Counter overflows			

16.5.15 SYSRTC_GRP0_IF - Group Interrupt Flags

Offset	Bit Position																															
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													CAP0	CMP1	CMP0	OVF

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	CAP0	0x0	RW	**Capture 0 Interrupt Flag**
	This bit is set when CAP0VALUE is updated			
2	CMP1	0x0	RW	**Compare 1 Interrupt Flag**
	This bit is set when counter matches COMP1VALUE			
1	CMP0	0x0	RW	**Compare 0 Interrupt Flag**
	This bit is set when counter matches COMP0VALUE			
0	OVF	0x0	RW	**Overflow Interrupt Flag**
	This bit is set when counter overflows			

16.5.16 SYSRTC_GRP0_IEN - Group Interrupt Enables

Offset	Bit Position																															
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													CAP0	CMP1	CMP0	OVF

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	CAP0	0x0	RW	**Capture 0 Interrupt Enable**
	Enable CAP0 interrupt			
2	CMP1	0x0	RW	**Compare 1 Interrupt Enable**
	Enable CMP1 interrupt			
1	CMP0	0x0	RW	**Compare 0 Interrupt Enable**
	Enable CMP0 interrupt			
0	OVF	0x0	RW	**Overflow Interrupt Enable**
	Enable OVF interrupt			

16.5.17 SYSRTC_GRP0_CTRL - Group Control Register

Offset	Bit Position																														
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																						0x0		0x0			0x0			0x0	0x0	0x0
Access																						RW		RW			RW			RW	RW	RW
Name																						CAP0EDGE		CMP1CMOA			CMP0CMOA			CAP0EN	CMP1EN	CMP0EN

Bit	Name	Reset	Access	Description
31:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:9	CAP0EDGE	0x0	RW	**Capture 0 Edge Select**
	Select edge detection for Capture 0			
	Value	Mode		Description
	0	RISING		Rising edges detected
	1	FALLING		Falling edges detected
	2	BOTH		Both edges detected
8:6	CMP1CMOA	0x0	RW	**Compare 1 Compare Match Output Action**
	Select PRS output action on Compare 1 match			
	Value	Mode		Description
	0	CLEAR		Cleared on the next cycle
	1	SET		Set on the next cycle
	2	PULSE		Set on the next cycle, cleared on the cycle after
	3	TOGGLE		Inverted on the next cycle
	4	CMPIF		Export this channel's CMP IF
5:3	CMP0CMOA	0x0	RW	**Compare 0 Compare Match Output Action**
	Select PRS output action on Compare 0 match			
	Value	Mode		Description
	0	CLEAR		Cleared on the next cycle
	1	SET		Set on the next cycle
	2	PULSE		Set on the next cycle, cleared on the cycle after
	3	TOGGLE		Inverted on the next cycle
	4	CMPIF		Export this channel's CMP IF
2	CAP0EN	0x0	RW	**Capture 0 Enable**
	Set this bit to enable Capture 0			
1	CMP1EN	0x0	RW	**Compare 1 Enable**
	Set this bit to enable Compare 1			

Bit	Name	Reset	Access	Description
0	CMP0EN	0x0	RW	**Compare 0 Enable**
	Set this bit to enable Compare 0			

16.5.18 SYSRTC_GRP0_CMP0VALUE - Compare 0 Value Register

Offset	Bit Position																													
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	CMP0VALUE																															

Bit	Name	Reset	Access	Description
31:0	CMP0VALUE	0x0	RW	**Compare 0 Value**
	Compare 0 match value			

16.5.19 SYSRTC_GRP0_CMP1VALUE - Compare 1 Value Register

Offset	Bit Position																													
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	CMP1VALUE																															

Bit	Name	Reset	Access	Description
31:0	CMP1VALUE	0x0	RW	**Compare 1 Value**
	Compare 1 match value			

16.5.20 SYSRTC_GRP0_CAP0VALUE - Capture 0 Value Register

Offset	Bit Position																															
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	CAP0VALUE																															

Bit	Name	Reset	Access	Description
31:0	CAP0VALUE	0x0	R	**Capture 0 Value**
	Capture 0 captured value			

16.5.21 SYSRTC_GRP0_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																															
0x068	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																										0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																										R	R	R	R	R	R	R
Name																										EMUACTIVE	HFXOACTIVE	EMUWAKEUP	HFXOSTART	CMP1VALUE	CMP0VALUE	CTRL

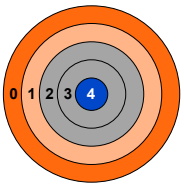
Bit	Name	Reset	Access	Description
31:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	EMUACTIVE	0x0	R	**Sync busy for EMUACTIVE bit**
	Last writing of EMUACTIVE is synchronizing to LF clock			
5	HFXOACTIVE	0x0	R	**Sync busy for HFXOACTIVE bit**
	Last writing of HFXOACTIVE is synchronizing to LF clock			
4	EMUWAKEUP	0x0	R	**Sync busy for EMUWAKEUP bit**
	Last writing of EMUWAKEUP is synchronizing to LF clock			
3	HFXOSTART	0x0	R	**Sync busy for HFXOSTART bit**
	Last writing of HFXOSTART is synchronizing to LF clock			
2	CMP1VALUE	0x0	R	**Sync busy for CMP1VALUE register**
	Last writing of CMP1VALUE is synchronizing to LF clock			
1	CMP0VALUE	0x0	R	**Sync busy for CMP0VALUE register**
	Last writing of CMP0VALUE is synchronizing to LF clock			
0	CTRL	0x0	R	**Sync busy for CTRL register**
	Last writing of CTRL is synchronizing to LF clock			

16.5.22 SYSRTC_GRP0_PRETRIG - Pretrigger Register

Offset	Bit Position																															
0x06C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0	0x0	0x0				0x0			
Access																							RW	RW	RW				RW			
Name																							EMUACTIVE	HFXOACTIVE	EMUWAKEUP				HFXOSTART			

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	EMUACTIVE	0x0	RW	**Status bit for EMU pretrigger**
	Status bit for EMU associated pretrigger			
8	HFXOACTIVE	0x0	RW	**Status bit for HFXO pretrigger**
	Status bit for HFXO associated pretrigger			
7:4	EMUWAKEUP	0x0	RW	**Cfg reg for Pretrigger event**
	Specifies number of clock cycles pretrigger event asserts before the compare irq			
3:0	HFXOSTART	0x0	RW	**Cfg reg for Pretrigger event**
	Specifies number of clock cycles pretrigger event asserts before the compare irq			

17. BURTC - Back-Up Real Time Counter



Quick Facts

What?

The BURTC is a 32 bit counter which operates on a low frequency oscillator and is capable of running in all energy modes.

Why?

It can provide periodic Wakeup events and PRS signals which can be used to wake up peripherals from any energy mode.

The availability of the BURTC in EM4, where most of the device is powered down, makes it ideal for keeping track of time in EM4.

How?

The BURTC provides a very wide range of periods for the interrupts facilitating flexible ultra-low energy operation.

17.1 Introduction

The Back-Up Real Time Counter (BURTC) is a 32-bit counter which operates on a low frequency oscillator, and is capable of running in all Energy Modes. It can provide periodic Wakeup events and PRS signals which can be used to wake up peripherals from any energy mode. The BURTC provides a very wide range of periods for the interrupts facilitating flexible ultra-low energy operation. The availability of the BURTC in EM4, where most of the device is powered down, makes it ideal for keeping track of time in EM4. A single compare channel is available which can be used to trigger an interrupt and/or wake the device up from a low energy mode.

17.2 Features

A low frequency oscillator is used as clock signal and the BURTC with one compare channel which can trigger wake-up, generate PRS signalling, or capture system events. 32-bit resolution and selectable prescaling allows the system to stay in low energy modes for long periods of time and still maintain reliable timekeeping.

- 32-bit Real Time Counter
- 15-bit pre-counter for flexible frequency scaling of main counter
- EM0 to EM4 operation and wakeup
- Reset only by External Pin and Power-On Resets
- Interrupt/wake up event after deterministic intervals
- PRS Outputs
- Debug mode
  - Configurable to either run or stop when processor is stopped (break)

17.3 Functional Description

An overview of the BURTC module is shown in Figure 17.1. BURTC Overview.

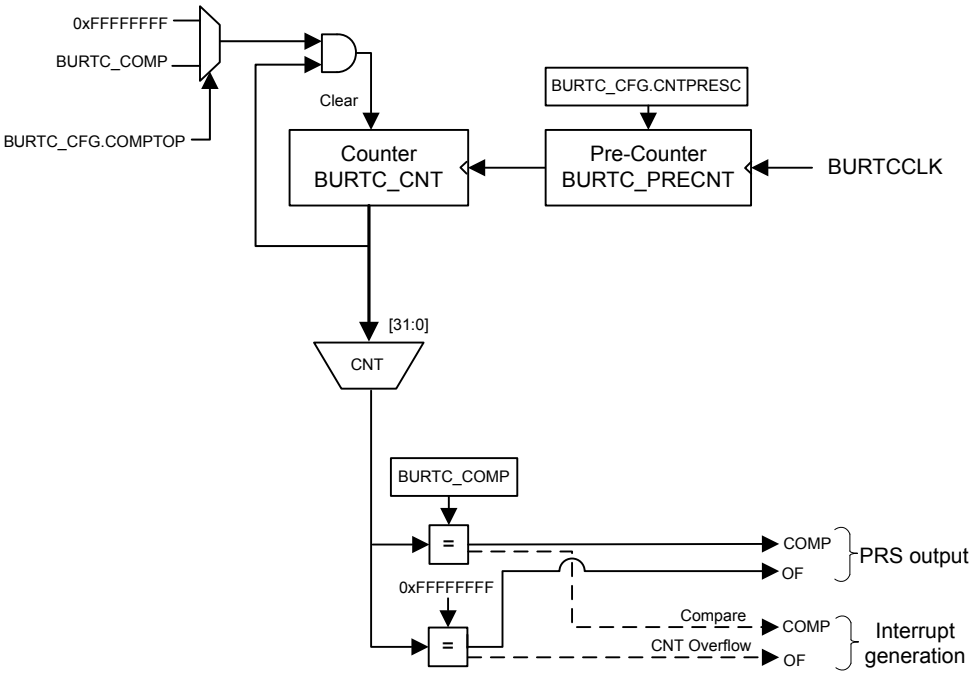


Figure 17.1. BURTC Overview

17.3.1 Clock Selection

The BURTC source clock (BURTCCLK) can be selected to be the LFXO, LFRCO, or ULFRCO by configuring the CMU_EM4GRPACLKCTRL.CLKSEL bitfield.

17.3.2 Configuration

To configure and use the BURTC properly, the following programming sequence must be followed:

1. Configure any desired options in the BURTC_CFG register. Note that the BURTC_CFG register can only be written when BURTC_EN.EN = 0 - a bus fault will occur if writing BURTC_CFG register while BURTC_EN.EN = 1.
2. Set BURTC_EN.EN = 1.
3. Set BURTC_CMD.START = 1 to start the BURTC counter.

   Note: All low frequency synchronization registers can only be programmed after EN is set to 1. The BURTC counter will only start to count once START command is issued. For HV Sync registers (e.g., BURTC_CMD), the first bitfield write will occur without issue. However, on subsequent bitfield writes to HV Sync registers, the firmware needs to poll the corresponding bit in BURTC_SYNCBUSY before programming the same bitfield once again.

To stop the BURTC, set BURTC_CMD.STOP = 1

17.3.3 Debug Features and Description

By default, the BURTC is halted when code execution is halted from the debugger. By setting the DEBUGRUN bit in the BURTC_CFG register, the BURTC will continue to run even when the debugger has halted the system.

17.3.4 Counter

The BURTC consists of two counters: the 32-bit main counter, BURTC_CNT, and a 15-bit pre-counter, BURTC_PRECNT. The pre-counter is a free running counter clocked by low frequency clock, used to generate a specific frequency for the main counter. The precounter will be counting only when the BURTC_CFG.CNTPRESC value is set greater than 0.

The BURTC peripheral clock is requested by setting the EN bit in BURTC_EN. Then the BURTC counters can be started by setting the command register START in BURTC_CMD. When BURTC_CMD.START has been initiated and BURTC_CFG.CNTPRESC > 0, the pre-counter (BURTC_PRECNT) increments upon each positive clock edge of the BURTCCLK, wrapping around to zero when it overflows.

The main counter can be accessed in BURTC_CNT register, and counts at frequency determined by the CNTPRESC bitfiled in BURTC_CFG. Setting CNTPRESC to 0 gives the maximum resolution, with the main counter clocked at the same frequency as the BURTCCLK. When CNTPRESC > 0, the main counter increments upon each tick given from the pre-counter, allowing the main counter ticks to be power-of-2 divisions of the BURTCCLK.

The Table 17.1. BURTC Resolution vs Overflow, $F_{BURTCCLK}$ = 32768 Hz table below shows the BURTC Resolution vs Overflow Time when using a 32768 Hz oscillator as the source clock of BURTC.

Table 17.1. BURTC Resolution vs Overflow, $F_{BURTCCLK}$ = 32768 Hz

BURTC_CFG.CNTPRESC	Main counter period, T_{CNT}	Overflow Time
DIV1	30.5 µs	36.4 hours
DIV2	61 µs	72.8 hours
DIV4	122 µs	145.6 hours
DIV8	244 µs	12 days
DIV16	488 µs	24 days
DIV32	977 µs	48 days
DIV64	1.95 ms	97 days
DIV128	3.91 ms	194 days
DIV256	7.81 ms	388 days
DIV512	15.6 ms	776 days
DIV1024	31.25 ms	4.2 years
DIV2048	62.5 ms	8.5 years
DIV4096	0.125 s	17 years
DIV8192	0.25 s	34 years
DIV16384	0.5 s	68 years
DIV32768	1 s	136 years

By default, the counter will keep counting until it reaches the top value, 0xFFFFFFFF, and then it wrap around and continue counting from zero. If COMPTOP in BURTC_CFG is set, the main counter will wrap to 0 on a Compare value match (i.e., BURTC_CNT = BURTC_COMP). If using the Compare value match, make sure to set COMPTOP prior to or at the same time the BURTC is enabled. Setting COMPTOP after enabling the BURTC will result in a bus fault error.

The counters of the BURTC, BURTC_CNT and BURTC_PRECNT, can at any time be written by software, as long as the registers are not locked using BURTC_LOCK. All BURTC control registers with Sync Type HV uses the 2 FF synchronization scheme.

Note: Writing to the BURTC_PRECNT register may alter the frequency of the ticks for the BURTC_CNT register.

17.3.5 Compare Channel

A single compare channel is available in the BURTC. The compare value is set in BURTC_COMP register. If BURTC_CFG.COMPTOP is set, the main counter will clear to 0 when it matches the value set in BURTC_COMP.

17.3.6 Interrupts

The BURTC has two interrupts: one for counter overflow and another for the compare match event. Individual interrupts are enabled by BURTC_IEN register bits. BURTC_EM4WUEN enables wakeup from EM4 for those events.

17.3.7 Register Lock

To prevent accidental writes to the BURTC registers, the BURTC_LOCK register can be written to any other value than the unlock value. To unlock the register, write the unlock value to BURTC_LOCK.LOCKKEY. Registers affected by this lock are:

- BURTC_CFG
- BURTC_EN
- BURTC_CMD
- BURTC_PRECNT
- BURTC_CNT
- BURTC_COMP
- BURTC_IEN

17.4 BURTC Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	BURTC_IPVERSION	R	IP Version ID
0x004	BURTC_EN	RW ENABLE	Module Enable Register
0x008	BURTC_CFG	RW CONFIG	Configuration Register
0x00C	BURTC_CMD	W LFSYNC	Command Register
0x010	BURTC_STATUS	RH	Status Register
0x014	BURTC_IF	RWH INTFLAG	Interrupt Flag Register
0x018	BURTC_IEN	RW	Interrupt Enable Register
0x01C	BURTC_PRECNT	RWH LFSYNC	Pre-Counter Value Register
0x020	BURTC_CNT	RWH LFSYNC	Counter Value Register
0x024	BURTC_EM4WUEN	RW	EM4 Wakeup Request Enable Register
0x028	BURTC_SYNCBUSY	RH	Synchronization Busy Register
0x02C	BURTC_LOCK	W	Configuration Lock Register
0x030	BURTC_COMP	RW LFSYNC	Compare Value Register
0x1000	BURTC_IPVERSION_SET	R	IP Version ID
0x1004	BURTC_EN_SET	RW ENABLE	Module Enable Register
0x1008	BURTC_CFG_SET	RW CONFIG	Configuration Register
0x100C	BURTC_CMD_SET	W LFSYNC	Command Register
0x1010	BURTC_STATUS_SET	RH	Status Register
0x1014	BURTC_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1018	BURTC_IEN_SET	RW	Interrupt Enable Register
0x101C	BURTC_PRECNT_SET	RWH LFSYNC	Pre-Counter Value Register
0x1020	BURTC_CNT_SET	RWH LFSYNC	Counter Value Register
0x1024	BURTC_EM4WUEN_SET	RW	EM4 Wakeup Request Enable Register
0x1028	BURTC_SYNCBUSY_SET	RH	Synchronization Busy Register
0x102C	BURTC_LOCK_SET	W	Configuration Lock Register
0x1030	BURTC_COMP_SET	RW LFSYNC	Compare Value Register
0x2000	BURTC_IPVERSION_CLR	R	IP Version ID
0x2004	BURTC_EN_CLR	RW ENABLE	Module Enable Register
0x2008	BURTC_CFG_CLR	RW CONFIG	Configuration Register
0x200C	BURTC_CMD_CLR	W LFSYNC	Command Register
0x2010	BURTC_STATUS_CLR	RH	Status Register
0x2014	BURTC_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2018	BURTC_IEN_CLR	RW	Interrupt Enable Register
0x201C	BURTC_PRECNT_CLR	RWH LFSYNC	Pre-Counter Value Register
0x2020	BURTC_CNT_CLR	RWH LFSYNC	Counter Value Register
0x2024	BURTC_EM4WUEN_CLR	RW	EM4 Wakeup Request Enable Register
0x2028	BURTC_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x202C	BURTC_LOCK_CLR	W	Configuration Lock Register
0x2030	BURTC_COMP_CLR	RW LFSYNC	Compare Value Register

Offset	Name	Type	Description
0x3000	BURTC_IPVERSION_TGL	R	IP Version ID
0x3004	BURTC_EN_TGL	RW ENABLE	Module Enable Register
0x3008	BURTC_CFG_TGL	RW CONFIG	Configuration Register
0x300C	BURTC_CMD_TGL	W LFSYNC	Command Register
0x3010	BURTC_STATUS_TGL	RH	Status Register
0x3014	BURTC_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3018	BURTC_IEN_TGL	RW	Interrupt Enable Register
0x301C	BURTC_PRECNT_TGL	RWH LFSYNC	Pre-Counter Value Register
0x3020	BURTC_CNT_TGL	RWH LFSYNC	Counter Value Register
0x3024	BURTC_EM4WUEN_TGL	RW	EM4 Wakeup Request Enable Register
0x3028	BURTC_SYNCBUSY_TGL	RH	Synchronization Busy Register
0x302C	BURTC_LOCK_TGL	W	Configuration Lock Register
0x3030	BURTC_COMP_TGL	RW LFSYNC	Compare Value Register

17.5 BURTC Register Description

17.5.1 BURTC_IPVERSION - IP Version ID

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP Version ID**

17.5.2 BURTC_EN - Module Enable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	When EN is cleared, DISABLING is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFOs. The CNT and PRECNT count registers are not reset during disablement.			
0	EN	0x0	RW	**BURTC Enable**
	Enable the BURTC to make the peripheral clock available to the module			

17.5.3 BURTC_CFG - Configuration Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0						0x0	0x0
Access																									RW						RW	RW
Name																									CNTPRESC						COMPTOP	DEBUGRUN

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:4	CNTPRESC	0x0	RW	**Counter prescaler value.** Configure counting frequency of the CNT register

Value	Mode	Description
0	DIV1	CLK_CNT = (BURTC LF CLK)/1
1	DIV2	CLK_CNT = (BURTC LF CLK)/2
2	DIV4	CLK_CNT = (BURTC LF CLK)/4
3	DIV8	CLK_CNT = (BURTC LF CLK)/8
4	DIV16	CLK_CNT = (BURTC LF CLK)/16
5	DIV32	CLK_CNT = (BURTC LF CLK)/32
6	DIV64	CLK_CNT = (BURTC LF CLK)/64
7	DIV128	CLK_CNT = (BURTC LF CLK)/128
8	DIV256	CLK_CNT = (BURTC LF CLK)/256
9	DIV512	CLK_CNT = (BURTC LF CLK)/512
10	DIV1024	CLK_CNT = (BURTC LF CLK)/1024
11	DIV2048	CLK_CNT = (BURTC LF CLK)/2048
12	DIV4096	CLK_CNT = (BURTC LF CLK)/4096
13	DIV8192	CLK_CNT = (BURTC LF CLK)/8192
14	DIV16384	CLK_CNT = (BURTC LF CLK)/16384
15	DIV32768	CLK_CNT = (BURTC LF CLK)/32768

Bit	Name	Reset	Access	Description
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	COMPTOP	0x0	RW	**Compare Channel is Top Value** When set, the counter is cleared in the clock cycle after a compare match with compare channel

Value	Mode	Description
0	DISABLE	The top value of the BURTC is 4294967295 (0xFFFFFFFF)
1	ENABLE	The top value of the BURTC is given by COMP

Bit	Name	Reset	Access	Description
0	DEBUGRUN	0x0	RW	**Debug Mode Run Enable** Set this bit to enable the BURTC to keep running in debug

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	X0		BURTC is frozen in debug mode
	1	X1		BURTC is running in debug mode

17.5.4 BURTC_CMD - Command Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															W(nB)	W(nB)
Name																															STOP	START

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	STOP	0x0	W(nB)	**Stop BURTC counter**
	Write a 1 to stop the BURTC counter.			
0	START	0x0	W(nB)	**Start BURTC counter**
	Write a 1 to start the BURTC counter.			

17.5.5 BURTC_STATUS - Status Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	R
Name																															LOCK	RUNNING

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	LOCK	0x0	R	**Configuration Lock Status**
	Indicates the current status of BURTC Lock			

Value	Mode	Description
0	UNLOCKED	All BURTC lockable registers are unlocked.
1	LOCKED	All BURTC lockable registers are locked.

0	RUNNING	0x0	R	**BURTC running status**
	Indicates the current status of BURTC running			

17.5.6 BURTC_IF - Interrupt Flag Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															COMP	OF

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	COMP	0x0	RW	**Compare Match Interrupt Flag**
	Set on a compare match between CNT and COMP.			
0	OF	0x0	RW	**Overflow Interrupt Flag**
	Set on a CNT value overflow.			

17.5.7 BURTC_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															COMP	OF

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	COMP	0x0	RW	**Compare Match Interrupt Flag**
	Set to enable the COMPIF Interrupt			
0	OF	0x0	RW	**Overflow Interrupt Flag**
	Set to enable the OFIF Interrupt			

17.5.8 BURTC_PRECNT - Pre-Counter Value Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																		0x0													
Access																		RW													
Name																		PRECNT													

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:0	PRECNT	0x0	RW	**Pre-Counter Value**
	Gives access to the Pre-counter value of the BURTC.			

17.5.9 BURTC_CNT - Counter Value Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	CNT																															

Bit	Name	Reset	Access	Description
31:0	CNT	0x0	RW	**Counter Value**
	Gives access to the counter value of the BURTC.			

17.5.10 BURTC_EM4WUEN - EM4 Wakeup Request Enable Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															COMPEM4WUEN	OFEM4WUEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	COMPEM4WUEN	0x0	RW	**Compare Match EM4 Wakeup Enable**
	Compare Match EM4 wakeup requests. No Synchronization done into peripheral clock domain.			
0	OFEM4WUEN	0x0	RW	**Overflow EM4 Wakeup Enable**
	Overflow EM4 Wakeup request. No Synchronization done into peripheral clock domain.			

17.5.11 BURTC_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												R	R	R	R	R
Name																												COMP	CNT	PRECNT	STOP	START

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	COMP	0x0	R	**Sync busy for COMP**
	Last writing of COMP is synchronizing to LF clock			
3	CNT	0x0	R	**Sync busy for CNT**
	Last writing of CNT is synchronizing to LF clock			
2	PRECNT	0x0	R	**Sync busy for PRECNT**
	Last writing of PRECNT is synchronizing to LF clock			
1	STOP	0x0	R	**Sync busy for STOP**
	Last writing of STOP is synchronizing to LF clock			
0	START	0x0	R	**Sync busy for START**
	Last writing of START is synchronizing to LF clock			

17.5.12 BURTC_LOCK - Configuration Lock Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0xAEE8														
Access																	W														
Name																	LOCKKEY														

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0xAEE8	W	**Configuration Lock Key**

Write any other value than the unlock code to lock BURTC_EN, BURTC_CFG, BURTC_CMD, BURTC_PRECNT, BURTC_CNT and BURTC_COMP registers from editing. Write the unlock code to unlock.

Value	Mode	Description
44776	UNLOCK	Write to unlock all BURTC lockable registers

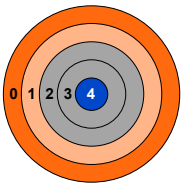
17.5.13 BURTC_COMP - Compare Value Register

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	COMP																															

Bit	Name	Reset	Access	Description
31:0	COMP	0x0	RW	**Compare Value**

A compare match event occurs when CNT is equal to this value. This event sets the COMP interrupt flag. It is also available as a PRS signal.

18. BURAM - Backup RAM



Quick Facts

What?

The BURAM is a dedicated 128-byte low-power RAM that is retained in EM4.

Why?

Most of the system, including the RAM, is powered off at EM4 entry to minimize current draw. The purpose of the BURAM is to retain critical data for use when the system wakes up.

How?

Because it is separate from the main system RAM, the BURAM has a dedicated power supply that is not shut down when the system enters EM4.

18.1 Introduction

The Back-Up RAM (BURAM) is a dedicated 128-byte RAM that remains powered when the system enters EM4. Upon exit from EM4, the data retained in the BURAM can be accessed by the application software.

18.2 Functional Description

The BURAM consists of 32 x 32-bit registers, which are retained in all energy modes, including EM4. Each word in the BURAM is accessible through the corresponding 32 BURAM_RETx_REG register. Note that each BURAM_RETx_REG register has an undefined state out of reset.

18.3 BURAM Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	BURAM_RETx_REG	RW	Retention Register
0x1000	BURAM_RETx_REG_SET	RW	Retention Register
0x2000	BURAM_RETx_REG_CLR	RW	Retention Register
0x3000	BURAM_RETx_REG_TGL	RW	Retention Register

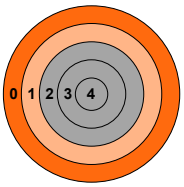
18.4 BURAM Register Description

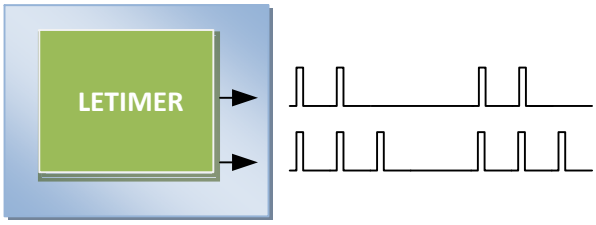
18.4.1 BURAM_RETx_REG - Retention Register

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW(nB)																															
Name	RETREG																															

Bit	Name	Reset	Access	Description
31:0	RETREG	0x0	RW(nB)	**Latch based Retention register**
	The RETREG registers are undefined out of reset. Any written RETREG values will be retained through any event other than a brownout or power-on reset.			

19. LETIMER - Low Energy Timer





Quick Facts

What?

The LETIMER is a down-counter that can keep track of time and output configurable waveforms. Running on a 32.768 Hz clock, the LETIMER is available in EM0 and EM1.

Why?

The LETIMER can be used to provide repeatable waveforms to external components while remaining in low-energy sleep modes. It is well-suited for applications such as metering systems or to provide more compare values than available in the SYSRTC.

How?

With buffered repeat and top value registers, the LETIMER can provide glitch-free waveforms at frequencies up to 16 kHz. It can be coupled with SYSRTC using PRS, allowing advanced time-keeping and wake-up functions in low-energy modes down to EM1.

19.1 Introduction

The LETIMER is a down-counter that can keep track of time and output configurable waveforms with minimal software intervention. Running on a Low Frequency clock, the LETIMER is available in EM0 and EM1. Because of this, it can be used for timing and output generation when most of the device is powered down, allowing simple tasks to be performed while the power consumption of the system is kept at an absolute minimum. It is well suited for applications such as metering systems or to provide more compare values than available in the SYSRTC. With buffered repeat and top value registers, the LETIMER can provide glitch-free waveforms at frequencies up to 16 kHz. It can be coupled with other peripherals using PRS, allowing advanced time-keeping and wake-up functions

19.2 Features

High-level features

- 24-bit Down counter
- 8-bit prescalar
- 2 Compare match registers
- TOP register can be Timer top value
- TOP register can be double buffered using TOPBUFF register
- Double buffered 8-bit Repeat Register
- Timer Start/Stop/Clear trigger can be from PRS or Software
- Configurable 2 Output pins - Toggle/Pulse/PWM
- Interrupt - Compare match/Timer underflow/Repeat done
- Optionally runs during debug
- 2 output pins can optionally be configured to provide different waveforms on timer underflow:
  - Toggle output pin
  - Pulse output with width of One Prescaled clock period
  - PWM
- 2 PRS Output

19.3 Functional Description

An overview of the LETIMER module is shown in Figure 19.1. LETIMER Overview. The LETIMER is a 24-bit down-counter with two compare registers, LETIMERn_COMP0 and LETIMERn_COMP1. The LETIMERn_TOP register can optionally act as a top value for the counter. The repeat counter LETIMERn_REP0 allows the timer to count a specified number of times before it stops. Both the LETIMERn_TOP and LETIMERn_REP0 registers can be double buffered by the LETIMERn_TOPBUFF and LETIMERn_REP1 registers to allow continuous operation. The timer can generate a single pin output, or two linked outputs.

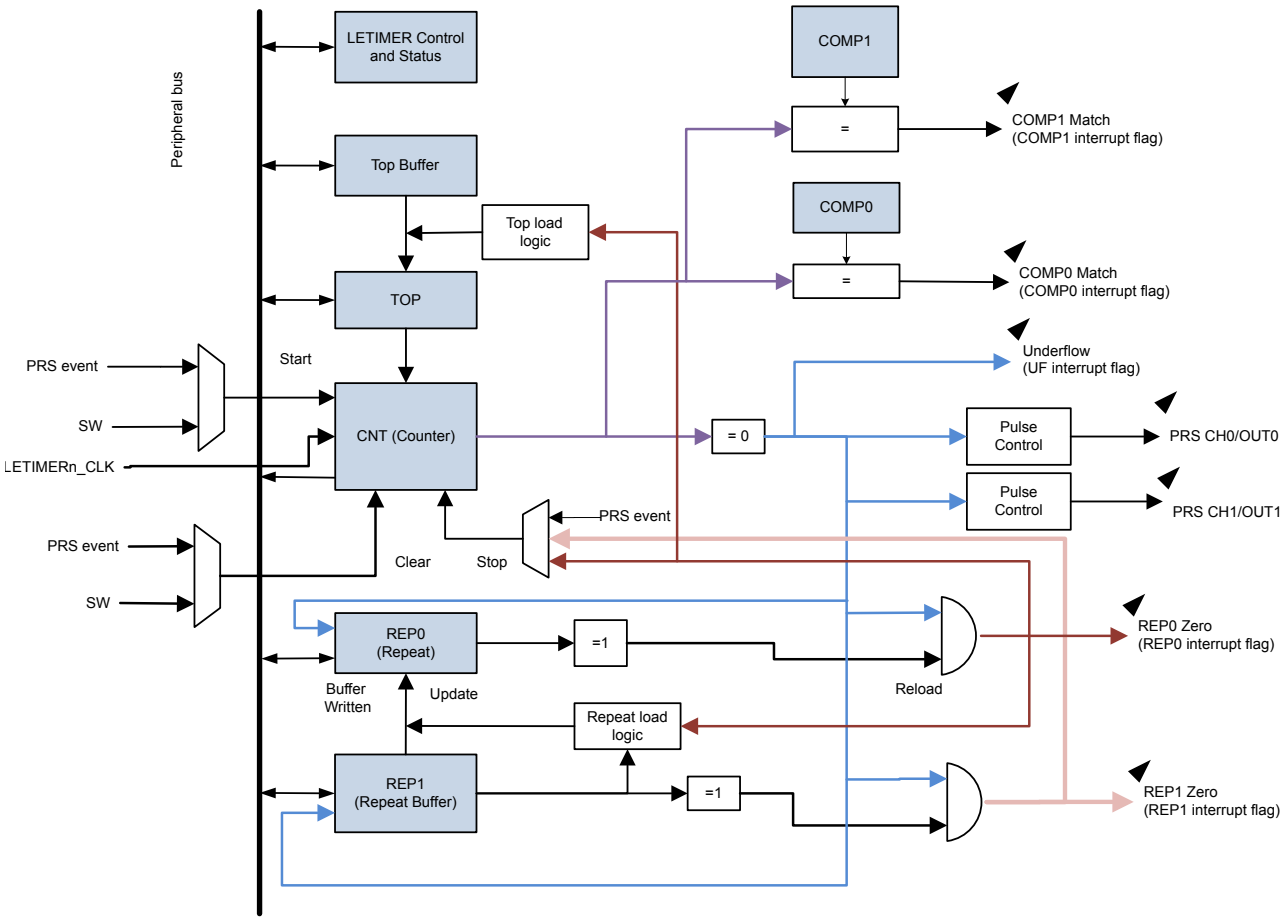


Figure 19.1. LETIMER Overview

19.3.1 Internal Overview

Timer

The timer value can be read using the LETIMERn_CNT register. The value can be written, and it can also be cleared by setting the CLEAR command bit in LETIMERn_CMD. If the CLEAR and START commands are issued at the same time, the timer will be cleared, then start counting at the top value.

Compare Registers

- The LETIMER has two compare match registers, LETIMERn_COMP0 and LETIMERn_COMP1. Each of these compare registers are capable of generating an interrupt when the counter value LETIMERn_CNT is equal to their value. When LETIMERn_CNT is equal to the value of LETIMERn_COMP0, the interrupt flag COMP0 in LETIMERn_IF is set, and when LETIMERn_CNT is equal to the value of LETIMERn_COMP1, the interrupt flag COMP1 in LETIMERn_IF is set.
- **Top Value**

  If CNTTOPEN in LETIMERn_CTRL is set, the value of LETIMERn_TOP acts as the top value of the timer, and LETIMERn_TOP is loaded into LETIMERn_CNT on timer underflow. If CNTTOPEN is cleared to 0, the timer wraps around to 0xFFFFFF. The underflow interrupt flag UF in LETIMERn_IF is set when the timer reaches zero.
- **Repeat Modes**

  By default, the timer wraps around to the top value or 0xFFFFFF on each underflow, and continues counting. The repeat counters can be used to get more control of the operation of the timer, including defining the number of times the counter should wrap around. Four different repeat modes are available, see Table 19.1. LETIMER Repeat Modes.

Table 19.1. LETIMER Repeat Modes

REPMODE	Mode	Description
0b00	Free-running	The timer runs until it is stopped.
0b01	One-shot	The timer runs as long as LETIMERn_REP0 != 0. LETIMERn_REP0 is decremented at each timer underflow.
0b10	Buffered	The timer runs as long as LETIMERn_REP0 != 0. LETIMERn_REP0 is decremented on each timer underflow. If LETIMERn_REP1 has been written with Non zero value, then it is loaded into LETIMERn_REP0 when LETIMERn_REP0 is about to be decremented to 0 and Timer countinue counting with new LETIMERn_REP0.
0b11	Double	The timer runs as long as LETIMERn_REP0 != 0 or LETIMERn_REP1 != 0. Both LETIMERn_REP0 and LETIMERn_REP1 are decremented at each timer underflow.

  The interrupt flags REP0 and REP1 in LETIMERn_IF are set whenever LETIMERn_REP0 or LETIMERn_REP1 are decremented to 0 respectively. REP0 is also set when the value of LETIMERn_REP1 is loaded into LETIMERn_REP0 in buffered mode.

  Write operations to LETIMERn_REP0 have priority over buffer loads from LETIMERn_REP1.
- **Buffered Top Value**

  In Buffered Mode, If BUFTOP in LETIMERn_CTRL is set, the value of LETIMERn_TOP is buffered by LETIMERn_TOPBUFF. In this mode, the value of LETIMERn_TOPBUFF is loaded into LETIMERn_TOP every time LETIMERn_REP0 is about to decrement to 0. This can be used to generate continually changing output waveforms.

  Write operations to LETIMERn_TOP have priority over buffer loads from LETIMERn_TOPBUFF.

19.3.2 Free Running Mode

In free-running mode, the LETIMER acts as a regular timer and the repeat operation is disabled. When started, the timer runs until it is stopped using the STOP command bit in LETIMERn_CMD/PRS. A state machine for this mode is shown in Figure 19.2. LETIMER State Machine for Free-running Mode .

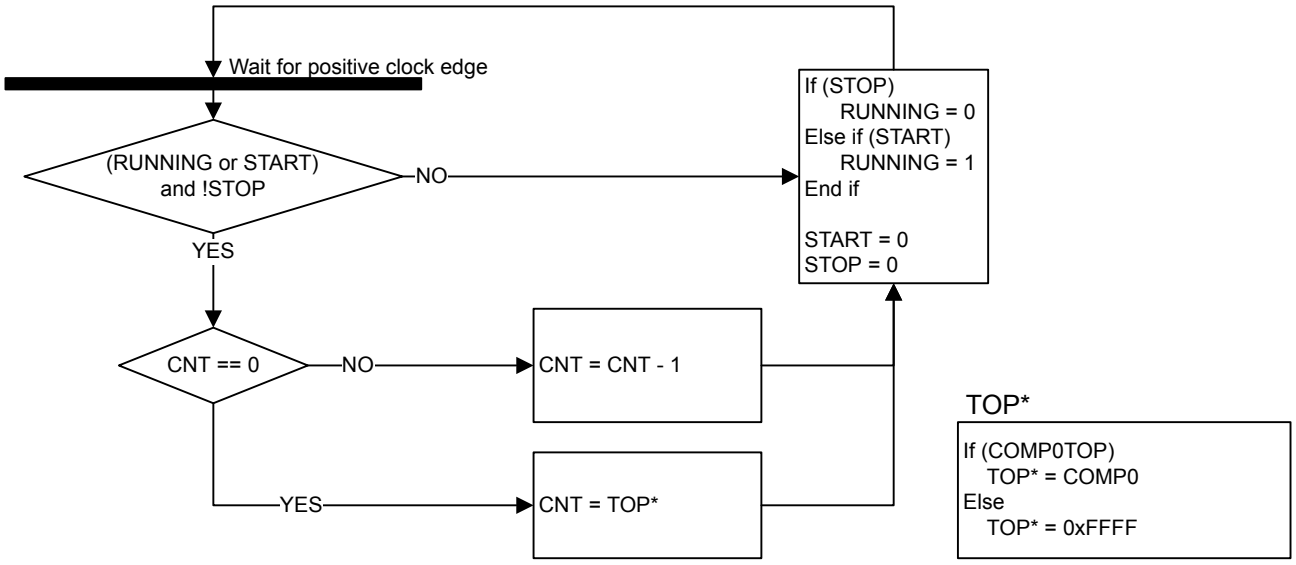


Figure 19.2. LETIMER State Machine for Free-running Mode

Note that the CLEAR command bit in LETIMERn_CMD always has priority over Decrement and Load TOP to LETIMERn_CNT. When the clear command is used, LETIMERn_CNT is set to 0 and an underflow event will not be generated when LETIMERn_CNT wraps around to the top value or 0xFFFFFF. Since no underflow event is generated, no output action is performed. LETIMERn_REP0, LETIMERn_REP1, LETIMERn_COMP0 and LETIMERn_COMP1 are also left untouched.

19.3.3 One-shot Mode

The one-shot repeat mode is the most basic repeat mode. In this mode, the repeat register LETIMERn_REP0 is decremented every time the timer underflows, and the timer stops when LETIMERn_REP0 goes from 1 to 0. In this mode, the timer counts down LETIMERn_REP0 times, i.e. the timer underflows LETIMERn_REP0 times.

Note: Note that write operations to LETIMERn_REP0 have priority over the timer decrement event. If LETIMERn_REP0 is assigned a new value in the same cycle as a timer decrement event occurs, the timer decrement will not occur and the new value is assigned.

LETIMERn_REP0 can be written while the timer is running to allow the timer to run for longer periods at a time without stopping. Write to LETIMERn_REP0 should be done after checking SYNC busy statusFigure 19.3. LETIMER One-shot Repeat State Machine .

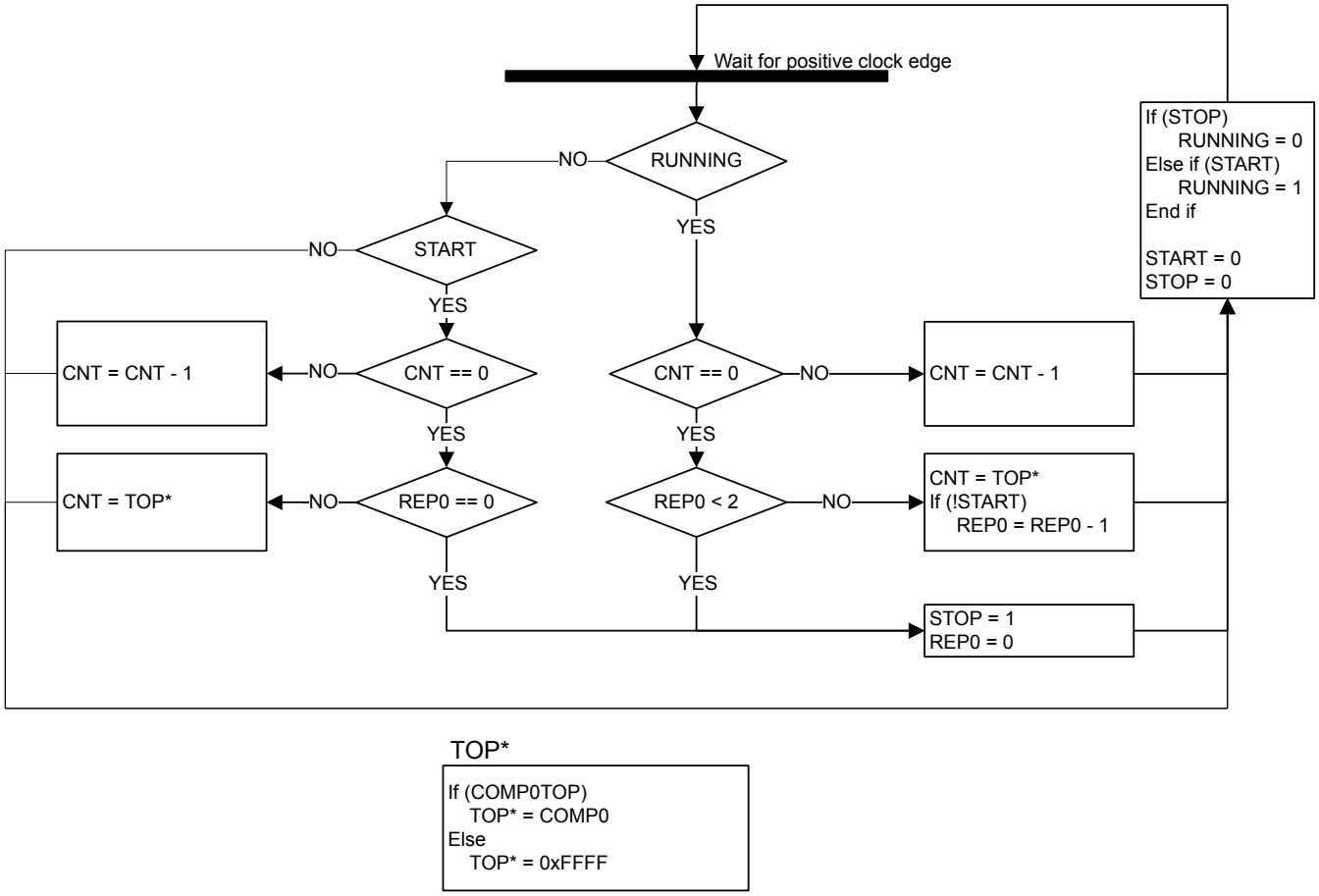


Figure 19.3. LETIMER One-shot Repeat State Machine

19.3.4 Buffered Mode

The Buffered repeat mode allows buffered timer operation. When started, the timer runs LETIMERn_REP0 number of times. If LETIMERn_REP1 has been written since the last time it was used and if it is nonzero, LETIMERn_REP1 is then loaded into LETIMERn_REP0, and counting continues the new number of times. The timer keeps going as long as LETIMERn_REP1 is updated with a nonzero value before LETIMERn_REP0 is finished counting down. The timer top value (LETIMERn_TOP) may also optionally be buffered using Top buff value (LETIMERn_TOPBUFF) by setting BUFTOP in LETIMERn_CTRL.

If the timer is started when both LETIMERn_CNT and LETIMERn_REP0 are zero but LETIMERn_REP1 is non-zero, LETIMERn_REP1 is loaded into LETIMERn_REP0, and the counter counts the loaded number of times.

Used in conjunction with a buffered top value, both the top and repeat values of the timer may be buffered, and the timer can for instance be set to run 4 times with period 7 (top value 6), 6 times with period 200, then 3 times with period 50.

A state machine for the buffered repeat mode is shown in Figure 19.4. LETIMER Buffered Repeat State Machine. $REP1_{USED}$ shown in the state machine is an internal variable that keeps track of whether the value in LETIMERn_REP1 has been loaded into LETIMERn_REP0 or not. The purpose of this is that a value written to LETIMERn_REP1 should only be counted once. $REP1_{USED}$ is cleared whenever LETIMERn_REP1 is used.

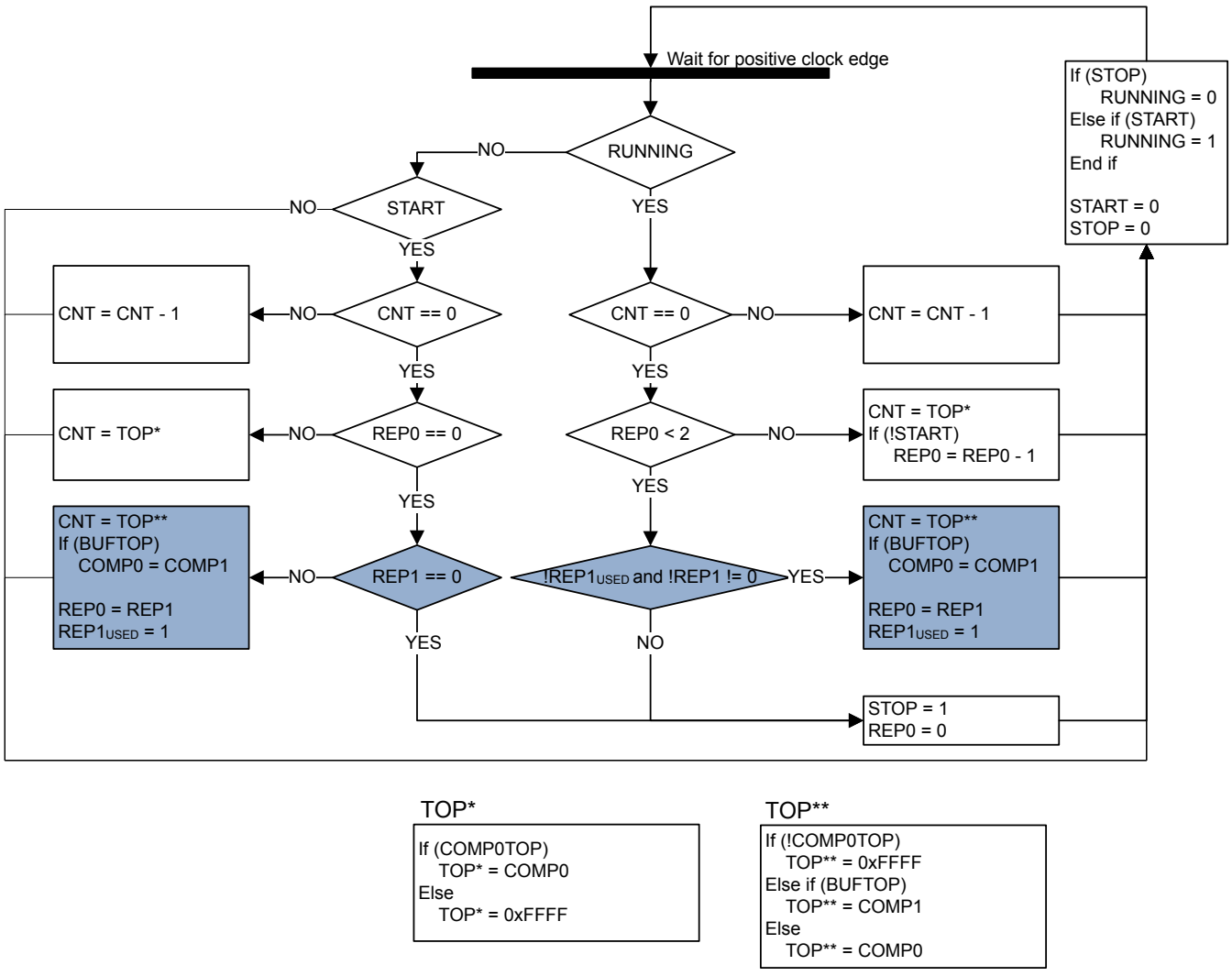


Figure 19.4. LETIMER Buffered Repeat State Machine

19.3.5 Double Mode

The Double repeat mode works much like the one-shot repeat mode. The difference is that, where the one-shot mode counts as long as LETIMERn_REP0 is larger than 0, the double mode counts as long as either LETIMERn_REP0 or LETIMERn_REP1 is larger than 0. As an example, say LETIMERn_REP0 is 3 and LETIMERn_REP1 is 10 when the timer is started. If no further interaction is done with the timer, LETIMERn_REP0 will now be decremented 3 times, and LETIMERn_REP1 will be decremented 10 times. The timer counts a total of 10 times, and LETIMERn_REP0 is 0 after the first three timer underflows and stays at 0. LETIMERn_REP0 and LETIMERn_REP1 can be written at any time. After a write to either of these, the timer is guaranteed to underflow at least the written number of times if the timer is running. Use the Double repeat mode to generate output on both the LETIMER outputs at the same time. The state machine for this repeat mode can be seen in Figure 19.5. LETIMER Double Repeat State Machine.

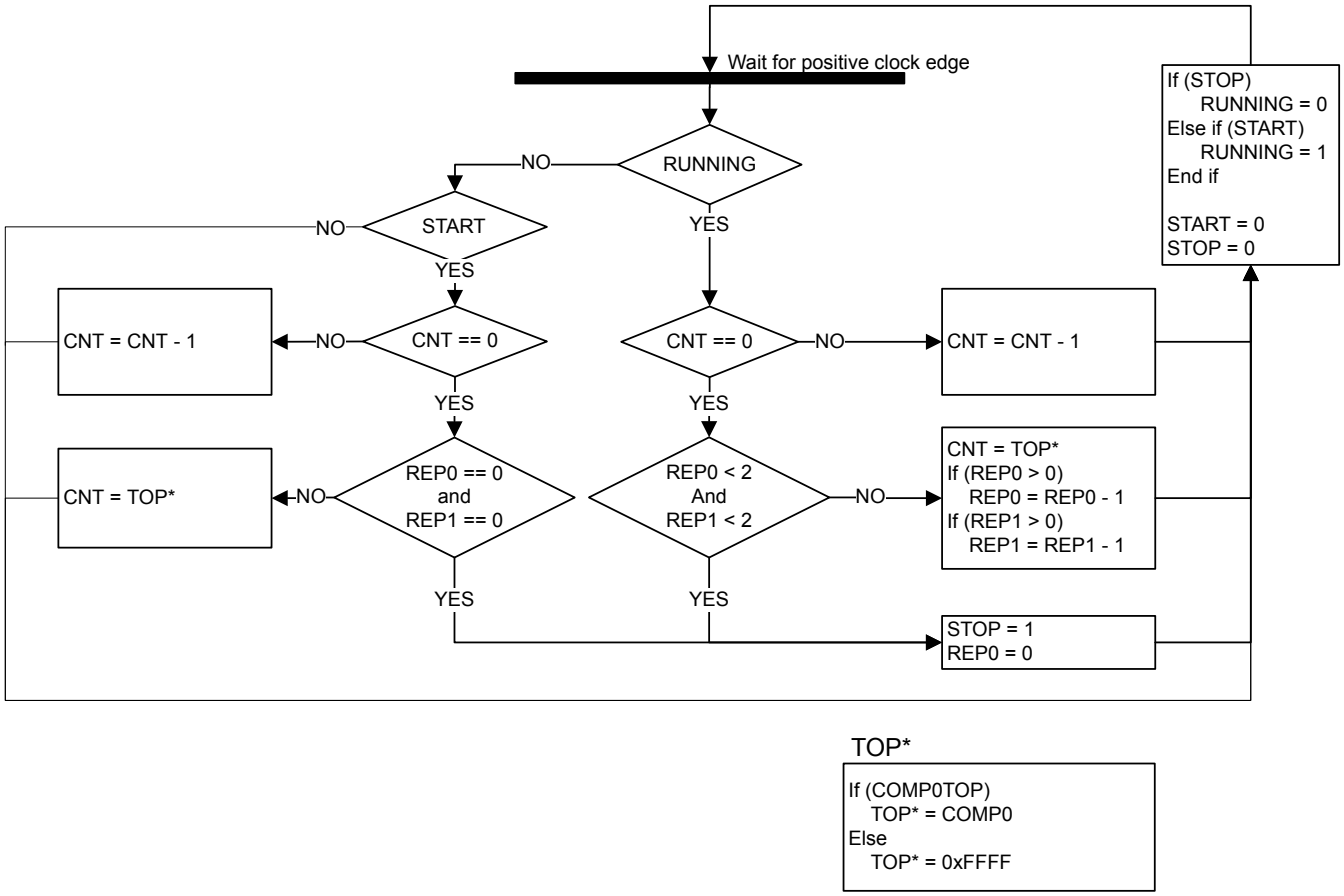


Figure 19.5. LETIMER Double Repeat State Machine

19.4 Clock Frequency

The LETIMER clock source is derived from EM23GRPACLK, which is selected in the Clock Management Unit (CMU), and is typically configured to have a frequency of 32 kHz from LFXO or LFRCO, or 1 kHz from ULFRCO. The LETIMER clock prescaler is defined by LETIMERn_CTRL.CNTPRESC.

The LETIMER Prescaled clock frequency is given by Figure 19.6. LETIMER Clock Frequency.

Clocked by LFRCO

$$f_{LETIMERn_CLK} = 32768/2^{CNTPRESC}$$

Clocked by ULFRCO

$$f_{LETIMERn_CLK} = 1024/2^{CNTPRESC}$$

Figure 19.6. LETIMER Clock Frequency

The exponent CNTPRESC is a 4 bit value in the LETIMERn_CTRL.CNTPRESC register bits.

To use this module, the LETIMERn_CLK must be enabled by writing 1 to LETIMERn_EN.EN.

19.5 PRS Input Triggers

The LETIMER can be configured to start, stop, and/or clear based on PRS inputs. The diagram showing the functions of the PRS input triggers is shown in Figure 19.7. LETIMER PRS input triggers..

There are 3 PRS inputs to the LETIMER, allowing the LETIMER to be started, stopped, or cleared based on the PRS inputs. The PRSSTARTMODE, PRSSTOPMODE, and PRSCLEARMODE bitfields in LETIMERn_PRSMODE select which edge or edge(s) will trigger the start, stop, and/or clear action.

The PRS channel inputs can be configured in the PRS_CONSUMER_LETIMERx_CLEAR, PRS_CONSUMER_LETIMERx_START, and PRS_CONSUMER_LETIMERx_STOP registers in the PRS module.

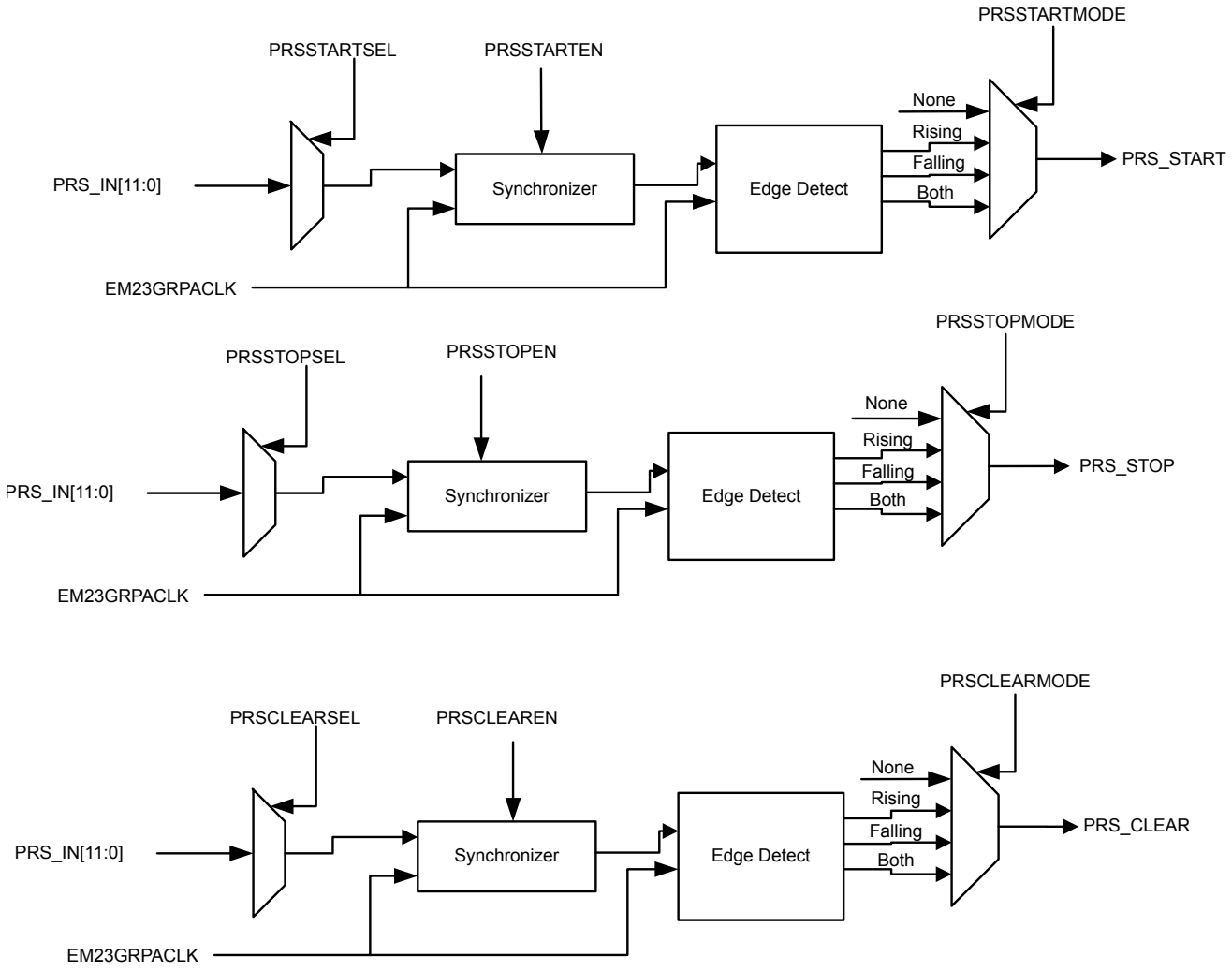


Figure 19.7. LETIMER PRS input triggers.

19.6 Debug

If DEBUGRUN in LETIMERn_CTRL is cleared, the LETIMER automatically stops counting when the CPU is halted during a debug session, and resumes operation when the CPU continues. Because of synchronization, the LETIMER is halted two clock cycles after the CPU is halted, and continues running two clock cycles after the CPU continues. RUNNING in LETIMERn_STATUS is not cleared when the LETIMER stops because of a debug-session.

Set DEBUGRUN in LETIMERn_CTRL to allow the LETIMER to continue counting even when the CPU is halted in debug mode.

19.7 Output Action

For each of the Outputs, an output action can be set.

The output actions can be set by configuring UFOA0 and UFOA1 in LETIMERn_CTRL. UFOA0 defines the action on output 0, while UFOA1 defines the action on output 1. The possible actions are defined in Table 19.2. LETIMER Underflow Output Actions.

Table 19.2. LETIMER Underflow Output Actions

UF0A0/UF0A1	Mode	Description
0b00	Idle	The output is held at its idle value
0b01	Toggle	The output is toggled on LETIMERn_CNT underflow
0b10	Pulse	The output is held active for one LF clock cycle on LETIMERn_CNT underflow It then returns to its idle value
0b11	PWM	The output is set idle on LETIMERn_CNT underflow and active on compare match with LETIMERn_COMP0/1.

Note: For the Pulse output Disabling LETIMER, Clearing Output while pulse output is generated can affect the pulse width.

Note: For Double mode, OUT0/1 generation is enabled when LETIMERn_REP0/1 != 0 respectively.

The polarity of the outputs can be set individually by configuring OPOL0 and OPOL1 in LETIMERn_CTRL. When these are cleared, their respective outputs have a low idle value and a high active value. When they are set, the idle value is high, and the active value is low. It is recommended to Clear outputs after changing polarity to makesure outputs take their default value.

When using the toggle action, the outputs can be driven to their idle values by setting their respective CTO0 or CTO1 command bits in LETIMERn_CMD. This can be used to put the output in a well-defined state before beginning to generate toggle output, which may be important in some applications. The command bit can also be used while the timer is running.

19.8 PRS Output

The LETIMER outputs can be routed out onto the PRS system. LETIMERn.OUT0 can be routed to PRS channel 0, and LETIMERn.OUT1 can be routed to PRS channel 1. Enabling the PRS connection can be done by setting SOURCESEL to LETIMERn and SIGSEL to CHx in PRS_ASYNC_CHx_CTRL.

19.9 Interrupts

The interrupts generated by the LETIMER are combined into one interrupt vector. If the interrupt for the LETIMER is enabled, an interrupt will be made if one or more of the interrupt flags in LETIMERn_IF and their corresponding bits in LETIMER_IEN are set.

19.10 Register Access

This module is a Low Energy Peripheral, and supports immediate synchronization. For description regarding immediate synchronization, refer to 4.2.4.4 Peripheral Access Performance.

Since this module is a Low Energy Peripheral, and runs off a clock which is asynchronous to the APB register clock, special considerations must be taken when accessing registers.

19.11 Programmer's Model

Important Note : Before writing any LFSYNC register, the module must be enabled (LETIMER_EN.EN) and the LETIMER_SYNCBUSY register should be polled to ensure the SYNC busy of that particular register field is not high.

Write LETIMER Configuration into LETIMER_CTRL Register

Enable clock to LETIMER module by setting LETIMER_EN.EN = 1

If used, write compare values into LETIMER_COMP0 and LETIMER_COMP1

If used, write repeat values into LETIMER_REP0 and LETIMER_REP1

If used, write LETIMER_TOP and LETIMER_TOPBUFF

If PRS is used as a trigger, configure LETIMER_PRSMODE accordingly

Enable Interrupts in LETIMER_IEN

Write LETIMER_CMD register to START Timer

19.11.1 Free Running Mode

LETIMER operation in Free running Mode with different output modes are shown in Figure 19.8. LETIMER - Free Running Mode Waveform. In this example, REPMODE in LETIMERn_CTRL is set to FREE, CNTTOPEN also in LETIMERn_CTRL has been set and LETIMERn_TOP has been written to 3. As seen in the figure, LETIMERn_TOP now decides the length of the signal periods. For the toggle mode, the period of the output signal is 2(LETIMERn_TOP + 1), and for the pulse modes, the periods of the output signals are LETIMERn_TOP+1. Note that the pulse outputs are delayed by one period relative to the toggle output. The pulses come at the end of their periods.

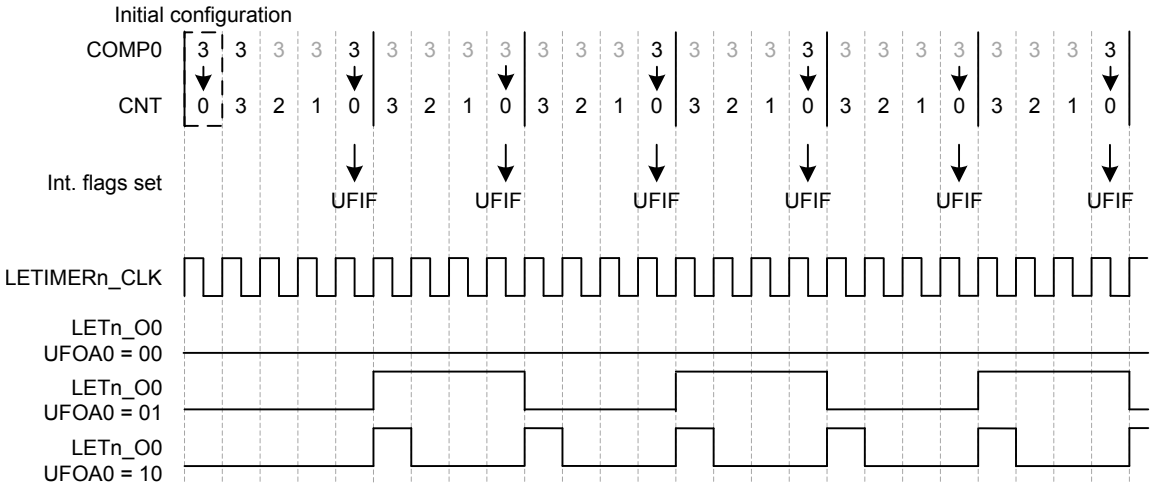


Figure 19.8. LETIMER - Free Running Mode Waveform

19.11.2 One Shot Mode

LETIMER operation in ONESHOT Mode with different output modes are shown in Figure 19.9. LETIMER - One Shot Mode Waveform. In this example, REPMODE in LETIMERn_CTRL is set to ONESHOT, CNTTOPEN also in LETIMERn_CTRL has been set and LETIMERn_TOP has been written to 3 and LETIMERn_REP0 has been written to 3. The resulting behavior is pretty similar to that shown in Figure 6, but in this case, the timer stops after counting to zero LETIMERn_REP0 times. By using LETIMERn_REP0 the user has full control of the number of pulses/toggles generated on the output.

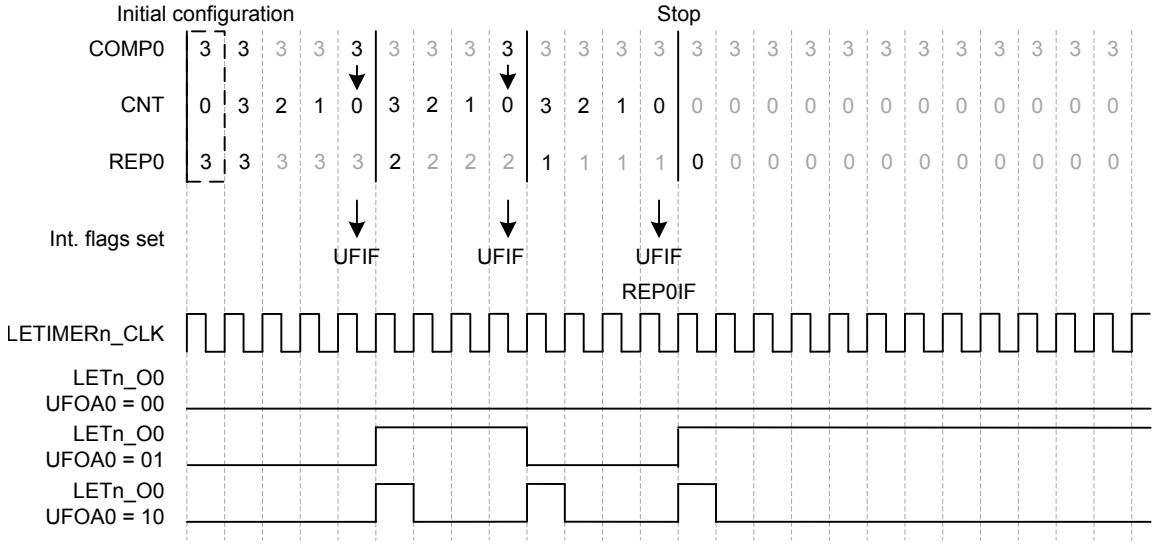


Figure 19.9. LETIMER - One Shot Mode Waveform

19.11.3 DOUBLE Mode

LETIMER operation in DOUBLE Mode with both outputs is shown in Figure 19.10. LETIMER - Double Mode Waveform. UFOA0 and UFOA1 in LETIMERn_CTRL are configured for pulse output and the outputs are configured for low idle polarity. As seen in the figure, the number written to the repeat registers determine the number of pulses generated on each of the outputs.

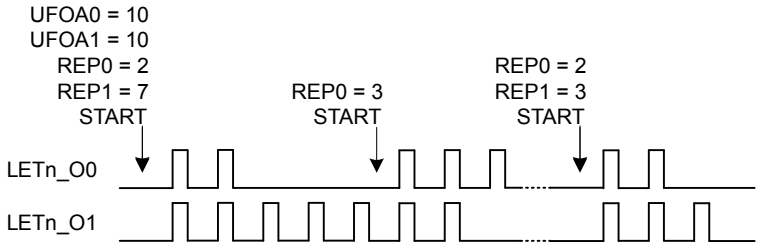


Figure 19.10. LETIMER - Double Mode Waveform

19.11.4 BUFFERED Mode

In BUFFERED Mode LETIMERn_TOPBUFF and LETIMERn_REP1 registers are used as Buffers for LETIMERn_TOP and LETIMERn_REP0 respectiverly. If both LETIMERn_TOP and LETIMERn_REP0 are 0 in buffered mode, and CNTTOPEN and BUFTOP in LETIMERn_CTRL are set, the values of LETIMERn_TOPBUFF and LETIMERn_REP1 are loaded into LETIMERn_TOP and LETIMERn_REP0 respectively when the timer is started. If no additional writes to LETIMERn_REP1 are done before the timer stops, LETIMERn_REP1 determines the number of pulses/toggles generated on the output, and LETIMERn_TOPBUFF determines the period lengths.

As the SYSRTC can also be used via PRS to start the LETIMER, the SYSRTC and LETIMER can thus be combined to generate specific pulse-trains at given intervals. Software can update LETIMERn_TOPBUFF and LETIMERn_REP1 to change the number of pulses and pulse-period in each train, but if changes are not required, software does not have to update the registers between each pulse train.

For the example in Figure 19.11. LETIMER - Buffered Mode Waveform, the initial values cause the LETIMER to generate two pulses with 3 cycle periods, or a single pulse 3 cycles wide every time the LETIMER is started. After the output has been generated, the LETIMER stops, and is ready to be triggered again.

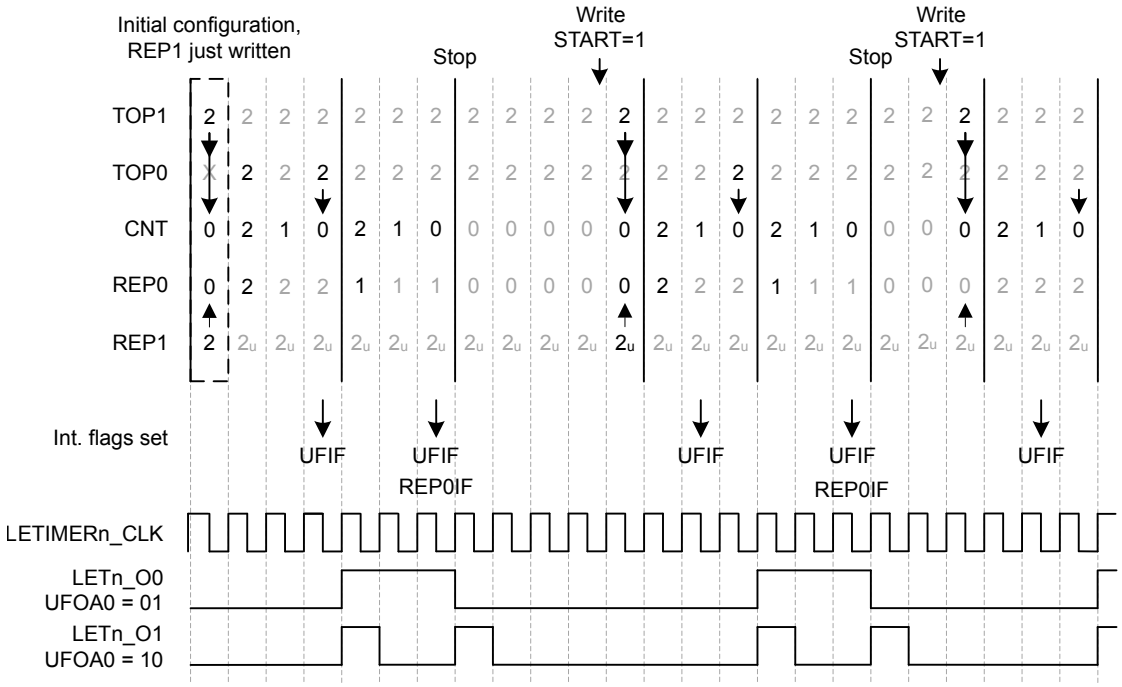


Figure 19.11. LETIMER - Buffered Mode Waveform

19.11.5 Continuous Output Generation

In some scenarios, it might be desired to make LETIMER generate a continuous waveform. Very simple constant waveforms can be generated without the repeat counter as shown in Figure 19.8. LETIMER - Free Running Mode Waveform, but to generate changing waveforms, using the repeat counter and buffer registers can prove advantageous.

For the example in Figure 19.12. LETIMER - Continuous Operation, the goal is to produce a pulse train consisting of 3 sequences with the following properties:

- 3 pulses with periods of 3 cycles
- 4 pulses with periods of 2 cycles
- 2 pulses with periods of 3 cycles

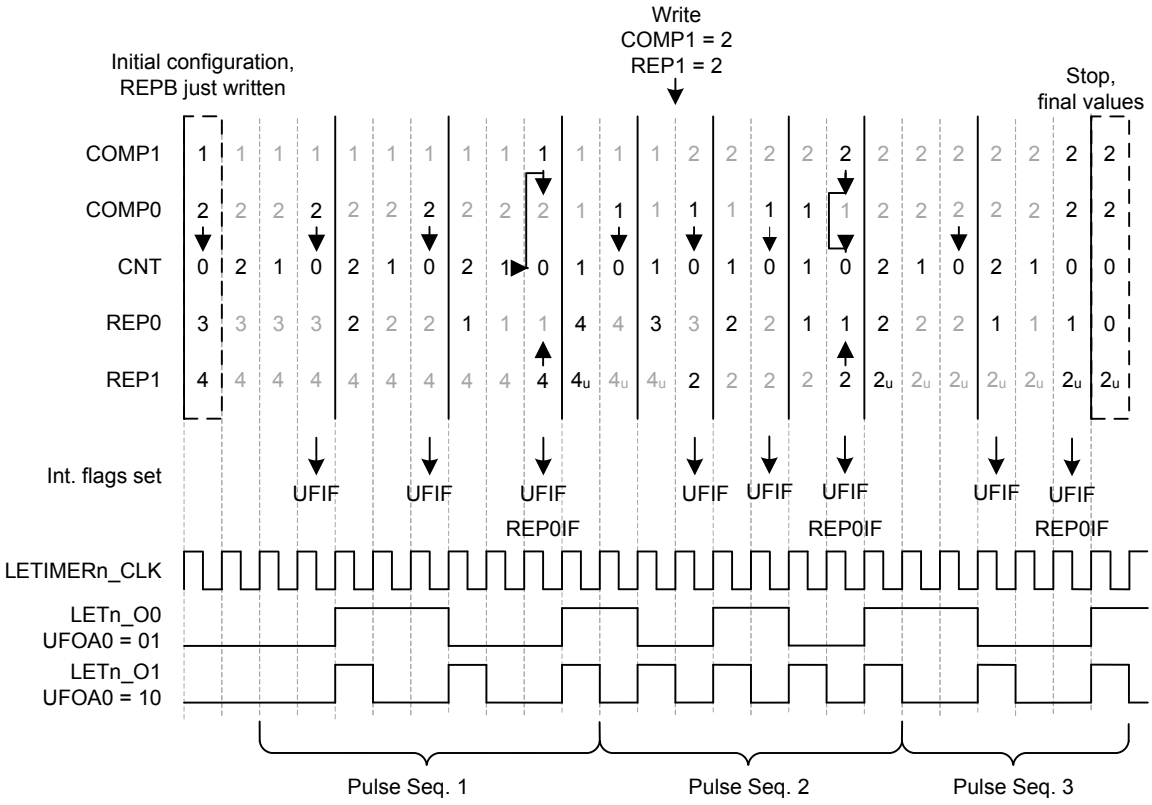


Figure 19.12. LETIMER - Continuous Operation

The first two sequences are loaded into the LETIMER before the timer is started.

LETIMERn_TOP is set to 2 (cycles – 1), and LETIMERn_REP0 is set to 3 for the first sequence, and the second sequence is loaded into the buffer registers, i.e. TOPBUFF is set to 1 and LETIMERn_REP1 is set to 4.

The LETIMER is set to trigger an interrupt when LETIMERn_REP0 is done by setting REP0 in LETIMERn_IEN. This interrupt is a good place to update the values of the buffers. Last but not least REPMODE in LETIMERn_CTRL is set to buffered mode, and the timer is started.

In the interrupt routine the buffers are updated with the values for the third sequence. If this had not been done, the timer would have stopped after the second sequence.

The final result is shown in Figure 19.12. LETIMER - Continuous Operation. The pulse output is grouped to show which sequence generated which output. Toggle output is also shown in the figure. Note that the toggle output is not aligned with the pulse outputs.

Note: Multiple LETIMER cycles are required to write a value to the LETIMER registers. The example in Figure 19.12. LETIMER - Continuous Operation assumes that writes are done in advance so they arrive in the LETIMER as described in the figure.

Figure 19.13. LETIMERn_CNT Not Initialized to 0 shows an example where the LETIMER is started while LETIMERn_CNT is nonzero. In this case the length of the first repetition is given by the value in LETIMERn_CNT.

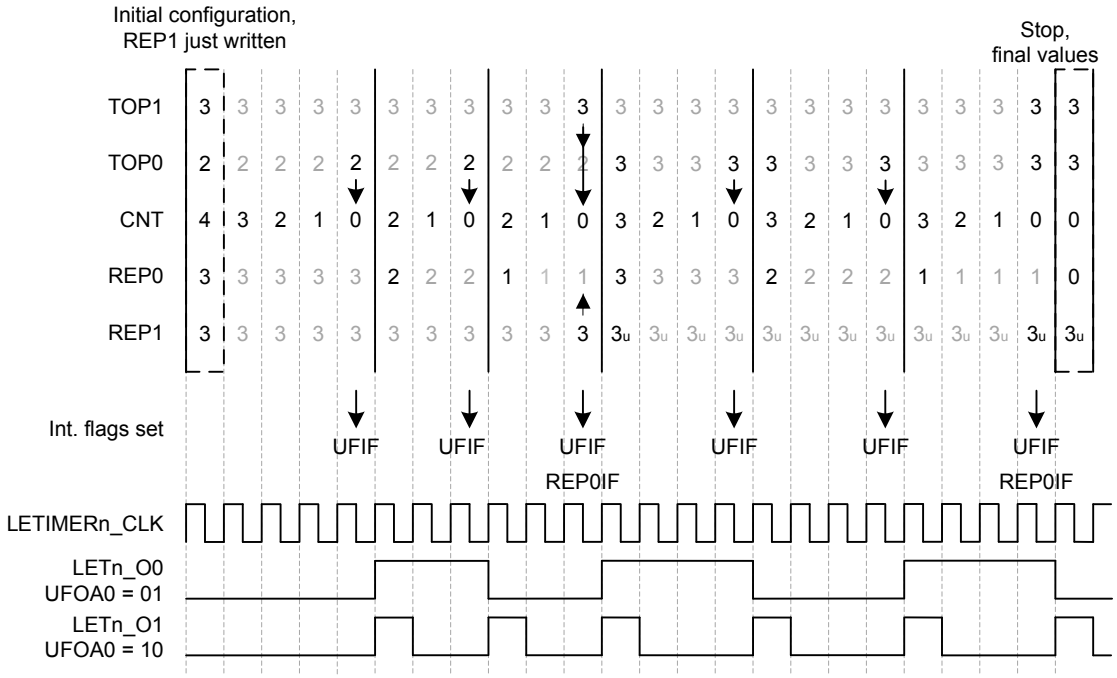


Figure 19.13. LETIMERn_CNT Not Initialized to 0

19.11.6 PWM Output

There are several ways of generating PWM output with the LETIMER, but the most straight-forward way is to use the PWM output mode. This mode is enabled by setting UFOA0 or UFOA1 in LETIMERn_CTRL to 3. In PWM mode, the output is set to idle on timer underflow, and active on LETIMERn_COMP0/1 match, so if for instance CNTTOPEN = 1 and OPOL0 = 0 in LETIMERn_CTRL, LETIMERn_TOP determines the PWM period, and LETIMERn_COMP0/1 determines the active period.

The PWM period in PWM mode is LETIMERn_TOP + 1. There is no special handling of the case where LETIMERn_COMP0/1 > LETIMERn_TOP, so if LETIMERn_COMP0/1 > LETIMERn_TOP, the PWM output is given by the idle output value. This means that for OPOLx = 0 in LETIMERn_CTRL, the PWM output will always be 0 for at least one clock cycle, and for OPOLx = 1 LETIMERn_CTRL, the PWM output will always be 1 for at least one clock cycle.

To generate a PWM signal using the full PWM range, invert OPOLx when LETIMERn_COMP0/1 is set to a value larger than LETIMERn_TOP.

19.12 LETIMER Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LETIMER_IPVERSION	R	IP Version
0x004	LETIMER_EN	RW ENABLE	Module Enable Register
0x008	LETIMER_SWRST	RW SWRST	Software Reset Register
0x00C	LETIMER_CTRL	RW	Control Register
0x010	LETIMER_CMD	W LFSYNC	Command Register
0x014	LETIMER_STATUS	RH	Status Register
0x018	LETIMER_CNT	RWH LFSYNC	Counter Value Register
0x01C	LETIMER_COMP0	RW	Compare Value Register 0
0x020	LETIMER_COMP1	RW	Compare Value Register 1
0x024	LETIMER_TOP	RWH LFSYNC	Counter TOP Value Register
0x028	LETIMER_TOPBUFF	RW	Buffered Counter TOP Value
0x02C	LETIMER_REP0	RWH LFSYNC	Repeat Counter Register 0
0x030	LETIMER_REP1	RWH LFSYNC	Repeat Counter Register 1
0x034	LETIMER_IF	RWH INTFLAG	Interrupt Flag Register
0x038	LETIMER_IEN	RW	Interrupt Enable Register
0x03C	LETIMER_LOCK	W	Configuration Lock Register
0x040	LETIMER_SYNCBUSY	RH	Synchronization Busy Register
0x050	LETIMER_PRSMODE	RW	PRS Input Mode Select Register
0x1000	LETIMER_IPVERSION_SET	R	IP Version
0x1004	LETIMER_EN_SET	RW ENABLE	Module Enable Register
0x1008	LETIMER_SWRST_SET	RW SWRST	Software Reset Register
0x100C	LETIMER_CTRL_SET	RW	Control Register
0x1010	LETIMER_CMD_SET	W LFSYNC	Command Register
0x1014	LETIMER_STATUS_SET	RH	Status Register
0x1018	LETIMER_CNT_SET	RWH LFSYNC	Counter Value Register
0x101C	LETIMER_COMP0_SET	RW	Compare Value Register 0
0x1020	LETIMER_COMP1_SET	RW	Compare Value Register 1
0x1024	LETIMER_TOP_SET	RWH LFSYNC	Counter TOP Value Register
0x1028	LETIMER_TOPBUFF_SET	RW	Buffered Counter TOP Value
0x102C	LETIMER_REP0_SET	RWH LFSYNC	Repeat Counter Register 0
0x1030	LETIMER_REP1_SET	RWH LFSYNC	Repeat Counter Register 1
0x1034	LETIMER_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1038	LETIMER_IEN_SET	RW	Interrupt Enable Register
0x103C	LETIMER_LOCK_SET	W	Configuration Lock Register
0x1040	LETIMER_SYNCBUSY_SET	RH	Synchronization Busy Register
0x1050	LETIMER_PRSMODE_SET	RW	PRS Input Mode Select Register
0x2000	LETIMER_IPVERSION_CLR	R	IP Version
0x2004	LETIMER_EN_CLR	RW ENABLE	Module Enable Register
0x2008	LETIMER_SWRST_CLR	RW SWRST	Software Reset Register

Offset	Name	Type	Description
0x200C	LETIMER_CTRL_CLR	RW	Control Register
0x2010	LETIMER_CMD_CLR	W LFSYNC	Command Register
0x2014	LETIMER_STATUS_CLR	RH	Status Register
0x2018	LETIMER_CNT_CLR	RWH LFSYNC	Counter Value Register
0x201C	LETIMER_COMP0_CLR	RW	Compare Value Register 0
0x2020	LETIMER_COMP1_CLR	RW	Compare Value Register 1
0x2024	LETIMER_TOP_CLR	RWH LFSYNC	Counter TOP Value Register
0x2028	LETIMER_TOPBUFF_CLR	RW	Buffered Counter TOP Value
0x202C	LETIMER_REP0_CLR	RWH LFSYNC	Repeat Counter Register 0
0x2030	LETIMER_REP1_CLR	RWH LFSYNC	Repeat Counter Register 1
0x2034	LETIMER_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2038	LETIMER_IEN_CLR	RW	Interrupt Enable Register
0x203C	LETIMER_LOCK_CLR	W	Configuration Lock Register
0x2040	LETIMER_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x2050	LETIMER_PRSMODE_CLR	RW	PRS Input Mode Select Register
0x3000	LETIMER_IPVERSION_TGL	R	IP Version
0x3004	LETIMER_EN_TGL	RW ENABLE	Module Enable Register
0x3008	LETIMER_SWRST_TGL	RW SWRST	Software Reset Register
0x300C	LETIMER_CTRL_TGL	RW	Control Register
0x3010	LETIMER_CMD_TGL	W LFSYNC	Command Register
0x3014	LETIMER_STATUS_TGL	RH	Status Register
0x3018	LETIMER_CNT_TGL	RWH LFSYNC	Counter Value Register
0x301C	LETIMER_COMP0_TGL	RW	Compare Value Register 0
0x3020	LETIMER_COMP1_TGL	RW	Compare Value Register 1
0x3024	LETIMER_TOP_TGL	RWH LFSYNC	Counter TOP Value Register
0x3028	LETIMER_TOPBUFF_TGL	RW	Buffered Counter TOP Value
0x302C	LETIMER_REP0_TGL	RWH LFSYNC	Repeat Counter Register 0
0x3030	LETIMER_REP1_TGL	RWH LFSYNC	Repeat Counter Register 1
0x3034	LETIMER_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3038	LETIMER_IEN_TGL	RW	Interrupt Enable Register
0x303C	LETIMER_LOCK_TGL	W	Configuration Lock Register
0x3040	LETIMER_SYNCBUSY_TGL	RH	Synchronization Busy Register
0x3050	LETIMER_PRSMODE_TGL	RW	PRS Input Mode Select Register

19.13 LETIMER Register Description

19.13.1 LETIMER_IPVERSION - IP Version

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP Version**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

19.13.2 LETIMER_EN - Module Enable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	When EN is cleared, DISABLING is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs, FIFOs etc.			
0	EN	0x0	RW	**module en**
	Enable the LETIMER module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit. When EN is cleared(disablement), it halts module operation immediately, and initialize the core domain such that when the is re-enabled, it starts cleanly.			

19.13.3 LETIMER_SWRST - Software Reset Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Software reset busy status** When SWRST command is issued, resetting logic sets RESETTING status immediately, and later it is cleared when reset process finishes.
0	SWRST	0x0	W	**Software reset command** A software reset command field resets the module back to the initial condition, similar to a power on reset condition

19.13.4 LETIMER_CTRL - Control Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0							0x0			0x0	0x0	0x0	0x0	0x0		0x0		0x0	
Access													RW							RW			RW	RW	RW	RW	RW		RW		RW	
Name													CNTPRESC							DEBUGRUN			CNTTOPEN	BUFTOP	OPOL1	OPOL0	UFOA1		UFOA0		REPMODE	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	CNTPRESC	0x0	RW	**Counter prescaler value**

Configure counting frequency of the CNT register. - Note - its not recommended to change this setting on the fly.

Value	Mode	Description
0	DIV1	CLK_CNT = (LETIMER LF CLK)/1
1	DIV2	CLK_CNT = (LETIMER LF CLK)/2
2	DIV4	CLK_CNT = (LETIMER LF CLK)/4
3	DIV8	CLK_CNT = (LETIMER LF CLK)/8
4	DIV16	CLK_CNT = (LETIMER LF CLK)/16
5	DIV32	CLK_CNT = (LETIMER LF CLK)/32
6	DIV64	CLK_CNT = (LETIMER LF CLK)/64
7	DIV128	CLK_CNT = (LETIMER LF CLK)/128
8	DIV256	CLK_CNT = (LETIMER LF CLK)/256

15:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	DEBUGRUN	0x0	RW	**Debug Mode Run Enable**

Set to keep the LETIMER running in debug mode.

Value	Mode	Description
0	DISABLE	LETIMER is frozen in debug mode
1	ENABLE	LETIMER is running in debug mode

11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	CNTTOPEN	0x0	RW	**Compare Value 0 Is Top Value**

When set, TOP value will be used as Counter Top Value

Value	Mode	Description
0	DISABLE	The top value of the LETIMER is 65535 (0xFFFF)
1	ENABLE	The top value of the LETIMER is given by COMP0

8	BUFTOP	0x0	RW	**Buffered Top**

Set to load TOPBUFF into TOP when REP0 reaches 0 in BUFFERED mode, allowing a buffered top value.

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	DISABLE		COMP0 is only written by software
	1	ENABLE		COMP0 is set to COMP1 when REP0 reaches 0
7	OPOL1	0x0	RW	**Output 1 Polarity**
	Defines the idle value of output 1.			
6	OPOL0	0x0	RW	**Output 0 Polarity**
	Defines the idle value of output 0.			
5:4	UFOA1	0x0	RW	**Underflow Output Action 1**
	Defines the action on OUT1 on a LETIMER underflow - IDLE/TOGGLE/PULSE/PWM			
	Value	Mode		Description
	0	NONE		LETIMERn_OUT1 is held at its idle value as defined by OPOL1
	1	TOGGLE		LETIMERn_OUT1 is toggled on CNT underflow
	2	PULSE		LETIMERn_OUT1 is held active for one LETIMER0 clock cycle on CNT underflow. The output then returns to its idle value as defined by OPOL1
	3	PWM		LETIMERn_OUT1 is set idle on CNT underflow, and active on compare match with COMP1
3:2	UFOA0	0x0	RW	**Underflow Output Action 0**
	Defines the action on OUT0 on a LETIMER underflow - IDLE/TOGGLE/PULSE/PWM			
	Value	Mode		Description
	0	NONE		LETIMERn_OUT0 is held at its idle value as defined by OPOL0
	1	TOGGLE		LETIMERn_OUT0 is toggled on CNT underflow
	2	PULSE		LETIMERn_OUT0 is held active for one LETIMER0 clock cycle on CNT underflow. The output then returns to its idle value as defined by OPOL0
	3	PWM		LETIMERn_OUT0 is set idle on CNT underflow, and active on compare match with COMP1
1:0	REPMODE	0x0	RW	**Repeat Mode**
	Repeat Mode - FREE/ONESHOT/BUFFERED/DOUBLE			
	Value	Mode		Description
	0	FREE		When started, the LETIMER counts down until it is stopped by software
	1	ONESHOT		The counter counts REP0 times. When REP0 reaches zero, the counter stops
	2	BUFFERED		The counter counts REP0 times. If REP1 has been written, it is loaded into REP0 when REP0 reaches zero, otherwise the counter stops
	3	DOUBLE		Both REP0 and REP1 are decremented when the LETIMER wraps around. The LETIMER counts until both REP0 and REP1 are zero

19.13.5 LETIMER_CMD - Command Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																												CTO1	CTO0	CLEAR	STOP	START

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	CTO1	0x0	W(nB)	**Clear Toggle Output 1**
	Set to drive toggle output 1 to its idle value			
3	CTO0	0x0	W(nB)	**Clear Toggle Output 0**
	Set to drive toggle output 0 to its idle value			
2	CLEAR	0x0	W(nB)	**Clear LETIMER**
	Set to clear LETIMER			
1	STOP	0x0	W(nB)	**Stop LETIMER**
	Set to stop LETIMER			
0	START	0x0	W(nB)	**Start LETIMER**
	Set to start LETIMER			

19.13.6 LETIMER_STATUS - Status Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	R
Name																															LETIMERLOCKSTATUS	RUNNING

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	LETIMERLOCKSTATUS	0x0	R	**LETIMER Lock Status**
	Indicates the current status of LETIMER Lock			

Value	Mode	Description
0	UNLOCKED	LETIMER registers are unlocked
1	LOCKED	LETIMER registers are locked

0	RUNNING	0x0	R	**LETIMER Running**
	Set when LETIMER is running.			

19.13.7 LETIMER_CNT - Counter Value Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																							
Access									RW																							
Name									CNT																							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	CNT	0x0	RW	**Counter Value**
	Use to read the current value of the LETIMER.			

19.13.8 LETIMER_COMP0 - Compare Value Register 0

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																							
Access									RW																							
Name									COMP0																							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	COMP0	0x0	RW	**Compare Value 0**
	Compare value for LETIMER.			

19.13.9 LETIMER_COMP1 - Compare Value Register 1

Offset	Bit Position																															
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																							
Access									RW																							
Name									COMP1																							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	COMP1	0x0	RW	**Compare Value 1**
	Compare and optionally buffered top value for LETIMER.			

19.13.10 LETIMER_TOP - Counter TOP Value Register

Offset	Bit Position																															
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																							
Access									RW																							
Name									TOP																							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	TOP	0x0	RW	**Counter TOP Value**
	TOP will be used as Counter TOP Value if CNTTOPEN is set to 1			

19.13.11 LETIMER_TOPBUFF - Buffered Counter TOP Value

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																						
Access									RW																						
Name									TOPBUFF																						

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:0	TOPBUFF	0x0	RW	**Buffered Counter TOP Value**
	TOPBUFF will be used as Counter TOP Value in BUFFERED Mode if CNTTOPEN and BUFFTOP is set set to 1			

19.13.12 LETIMER_REP0 - Repeat Counter Register 0

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									REP0							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	REP0	0x0	RW	**Repeat Counter 0**
	Optional repeat counter.			

19.13.13 LETIMER_REP1 - Repeat Counter Register 1

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									REP1							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	REP1	0x0	RW	**Repeat Counter 1**
	Optional repeat counter or buffer for REP0.			

19.13.14 LETIMER_IF - Interrupt Flag Register

Offset	Bit Position																														
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												REP1	REP0	UF	COMP1	COMP0

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	REP1	0x0	RW	**Repeat Counter 1 Interrupt Flag**
	Set when repeat counter 1 reaches zero.			
3	REP0	0x0	RW	**Repeat Counter 0 Interrupt Flag**
	Set when repeat counter 0 reaches zero or when the REP1 interrupt flag is loaded into the REP0 interrupt flag.			
2	UF	0x0	RW	**Underflow Interrupt Flag**
	Set on LETIMER underflow.			
1	COMP1	0x0	RW	**Compare Match 1 Interrupt Flag**
	Set when LETIMER reaches the value of COMP1.			
0	COMP0	0x0	RW	**Compare Match 0 Interrupt Flag**
	Set when LETIMER reaches the value of COMP0.			

19.13.15 LETIMER_IEN - Interrupt Enable Register

Offset	Bit Position																															
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												REP1	REP0	UF	COMP1	COMP0

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	REP1	0x0	RW	**Repeat Counter 1 Interrupt Enable**
	Repeat Counter 1 Interrupt Enable			
3	REP0	0x0	RW	**Repeat Counter 0 Interrupt Enable**
	Repeat Counter 0 Interrupt Enable			
2	UF	0x0	RW	**Underflow Interrupt Enable**
	Underflow Interrupt Enable			
1	COMP1	0x0	RW	**Compare Match 1 Interrupt Enable**
	Compare Match 1 Interrupt Enable			
0	COMP0	0x0	RW	**Compare Match 0 Interrupt Enable**
	Compare Match 0 Interrupt Enable			

19.13.16 LETIMER_LOCK - Configuration Lock Register

Offset	Bit Position																															
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	W															
Name																	LETIMERLOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LETIMERLOCKKEY	0x0	W	**Configuration Lock Key**
	Write any other value than the unlock code to lock LETIMER_EN, LETIMER_SWRST, LETIMER_CTRL, LETIMER_CMD, LETIMER_CNT, LETIMER_COMP0, LETIMER_COMP1, LETIMER_TOP, LETIMER_TOPBUFF, LETIMER_REP0, LETIMER_REP1 and PRSMODE registers from editing. Write the unlock code to unlock.			

Value	Mode	Description
52476	UNLOCK	Write to unock LETIMER lockable registers

19.13.17 LETIMER_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0		0x0
Access																							R	R	R	R	R	R	R	R		R
Name																							CTO1	CTO0	CLEAR	STOP	START	REP1	REP0	TOP		CNT

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	CTO1 Sync busy for CTO1	0x0	R	**Sync busy for CTO1**
8	CTO0 Sync busy for CTO0	0x0	R	**Sync busy for CTO0**
7	CLEAR Sync busy for CLEAR	0x0	R	**Sync busy for CLEAR**
6	STOP Sync busy for STOP	0x0	R	**Sync busy for STOP**
5	START Sync busy for START	0x0	R	**Sync busy for START**
4	REP1 Sync busy for REP1	0x0	R	**Sync busy for REP1**
3	REP0 Sync busy for REP0	0x0	R	**Sync busy for REP0**
2	TOP Sync busy for TOP	0x0	R	**Sync busy for TOP**
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CNT Sync busy for CNT	0x0	R	**Sync busy for CNT**

19.13.18 LETIMER_PRSMODE - PRS Input Mode Select Register

Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0				0x0				0x0																			
Access					RW				RW				RW																			
Name					PRSCLEARMODE				PRSSTOPMODE				PRSSTARTMODE																			

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:26	PRSCLEARMODE	0x0	RW	**PRS Clear Mode**
	Mode-NONE/RISING/FALLING/BOTH			

Value	Mode	Description
0	NONE	PRS cannot clear the LETIMER
1	RISING	Rising edge of selected PRS input can clear the LETIMER
2	FALLING	Falling edge of selected PRS input can clear the LETIMER
3	BOTH	Both the rising or falling edge of the selected PRS input can clear the LETIMER

Bit	Name	Reset	Access	Description
25:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:22	PRSSTOPMODE	0x0	RW	**PRS Stop Mode**
	Mode-NONE/RISING/FALLING/BOTH			

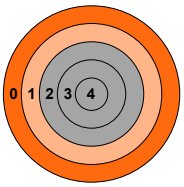
Value	Mode	Description
0	NONE	PRS cannot stop the LETIMER
1	RISING	Rising edge of selected PRS input can stop the LETIMER
2	FALLING	Falling edge of selected PRS input can stop the LETIMER
3	BOTH	Both the rising or falling edge of the selected PRS input can stop the LETIMER

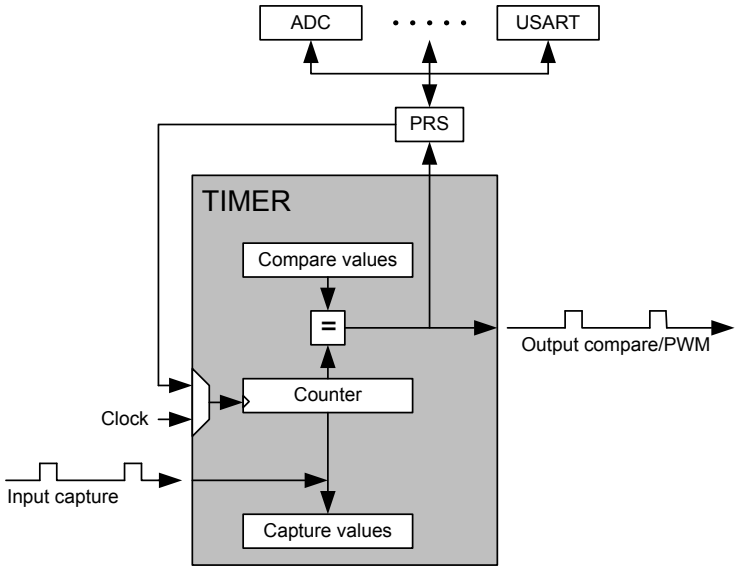
Bit	Name	Reset	Access	Description
21:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:18	PRSSTARTMODE	0x0	RW	**PRS Start Mode**
	Mode-NONE/RISING/FALLING/BOTH			

Value	Mode	Description
0	NONE	PRS cannot start the LETIMER
1	RISING	Rising edge of selected PRS input can start the LETIMER
2	FALLING	Falling edge of selected PRS input can start the LETIMER
3	BOTH	Both the rising or falling edge of the selected PRS input can start the LETIMER

Bit	Name	Reset	Access	Description
17:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

20. TIMER - Timer/Counter





Quick Facts

What?

The TIMER keeps track of timing and counts events, generates output waveforms, and triggers timed actions in other peripherals.

Why?

Most applications have activities that need to be timed accurately with as little CPU intervention and energy consumption as possible.

How?

The flexible 32-bit timer can be configured to provide PWM waveforms with optional dead-time insertion (e.g., motor control) or work as a frequency generator. The timer can also count events and control other peripherals through the PRS, which offloads the CPU and reduces energy consumption.

20.1 Introduction

The general purpose timer has up to 7 compare/capture channels for input capture and compare/Pulse-Width Modulation (PWM) output.

The TIMER module is 32 bits wide, and includes a three-channel Dead-Time Insertion module suitable for motor control applications.

Refer to the device data sheet to determine the capabilities of each timer instance.

20.2 Features

- 32-bit auto reload up/down counter
  - Dedicated 32-bit reload register which serves as counter maximum
- 3 to 7 Compare/Capture channels
  - Individually configurable as either input capture or output compare/PWM
- Multiple Counter modes
  - Count up
  - Count down
  - Count up/down
  - Quadrature Decoder
  - Direction and count from external pins
- 2x Count Mode
- Counter control from PRS or external pin
  - Start
  - Stop
  - Reload and start
- Inter-Timer connection
  - Allows 64-bit counter mode
  - Start/stop synchronization between several timers
- Input Capture
  - Period measurement
  - Pulse width measurement
  - 2-entry Input Capture FIFO for each capture channel
    - Capture on either positive or negative edge
    - Capture on both edges
  - Optional digital noise filtering on capture inputs
- Output Compare
  - Compare output toggle/pulse on compare match
  - Immediate update of compare registers
- PWM
  - Up-count PWM
  - Up/down-count PWM
  - Predictable initial PWM output state (configured by SW)
  - Buffered compare register to ensure glitch-free update of compare values
  - Per-channel programmable phase (up-count PWM mode)
  - Output dithering with configurable width (up-count PWM mode)
  - Output re-timing to mitigate RF interference
- Clock sources
  - EM01GRPACLK
    - 10-bit Prescaler
  - External pin
  - Peripheral Reflex System
- Debug mode
  - Configurable to either run or stop when processor is stopped (halt/breakpoint)
- Interrupts, PRS output and/or DMA request on:
  - Underflow
  - Overflow
  - Compare/Capture event

- Dead-Time Insertion Unit
  - Complementary PWM outputs with programmable dead-time
    - Dead-time is specified independently for rising and falling edge
      - 10-bit prescaler
      - 6-bit time value
    - Outputs have configurable polarity
    - Outputs can be set inactive individually by software.
  - Configurable action on fault
    - Set outputs inactive
    - Clear output
    - Tristate output
  - Individual fault sources
    - One or two PRS signals
    - Debugger
      - Support for automatic restart
    - Core lockup
  - Configuration lock

20.3 Functional Description

An overview of the TIMER module is shown in Figure 20.1. TIMER Block Overview and it consists of a 32 bit up/down counter with multiple compare/capture channels connected to pins TIMERn.CC0, TIMERn.CC1, and TIMERn.CC2. Three channels are shown in the diagram, but individual TIMER instances may support more channels.

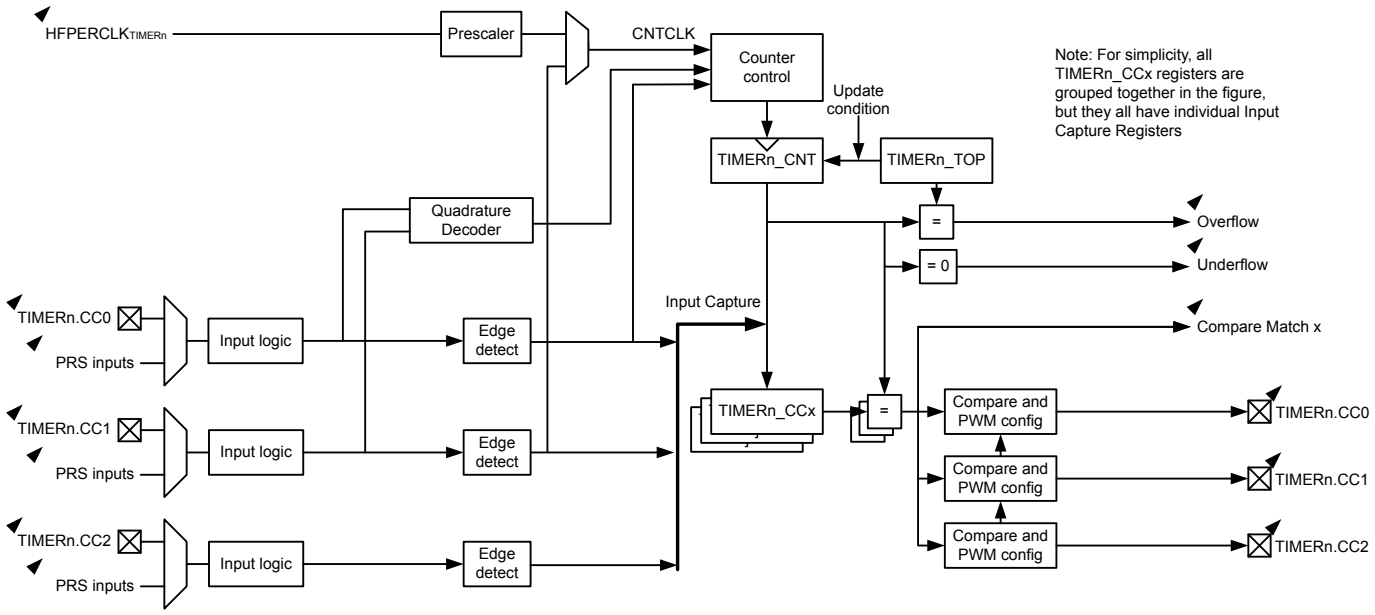


Figure 20.1. TIMER Block Overview

20.3.1 Register Access

The timer module interface consists of multiple register types. Registers of type "RW CONFIG" should only be written when the module is disabled (TIMERn_EN.EN = 0). Registers of type "W SYNC", "R SYNC" or "RW SYNC" should only be read or written when the module is enabled (TIMERn_EN.EN = 1). A typical setup sequence for a TIMER module is as follows:

1. With the TIMER disabled (TIMERn_EN.EN = 0), program any CONFIG registers required for the application.
2. Enable the TIMER by setting EN in TIMERn_EN to 1.
3. Program any non-CONFIG registers required for the application.
4. The TIMER is then ready for use.

20.3.2 Counter Modes

The timer consists of a counter that can be configured to the following modes, using the MODE field in TIMERn_CFG:

- Up-count: Counter counts up until it reaches the value in TIMERn_TOP, where it is reset to 0 before counting up again.
- Down-count: The counter starts at the value in TIMERn_TOP and counts down. When it reaches 0, it is reloaded with the value in TIMERn_TOP.
- Up/Down-count: The counter starts at 0 and counts up. When it reaches the value in TIMERn_TOP, it counts down until it reaches 0 and starts counting up again.
- Quadrature Decoder: Two input channels where one determines the count direction, while the other pin triggers a clock event.

In addition to the TIMER modes listed above, the TIMER also supports a 2x count mode. In this mode the counter increments/decrements by 2 on each clock edge. The 2x count mode can be used to double the PWM frequency when the compare/capture channel is put into PWM mode. The 2x count mode is enabled by setting the X2CNT bitfield in the TIMERn_CTRL register.

The counter value can be read or written by software any time the module is enabled by accessing the CNT field in TIMERn_CNT.

20.3.2.1 Events

The main counter can generate overflow and underflow events during operation.

Overflow (TIMERn_IF.OF) is set when the counter value shifts from TIMERn_TOP to the next value when counting up. In up-count mode and quadrature decoder mode the next value is 0. In up/down-count mode, the next value is TIMERn_TOP-1.

Underflow (TIMERn_IF.UF) is set when the counter value shifts from 0 to the next value when counting down. In down-count mode and quadrature decoder mode, the next value is TIMERn_TOP. In up/down-count mode the next value is 1.

An update event occurs on overflow in up-count mode and on underflow in down-count or up/down count mode. Additionally, an update event also occurs on overflow and underflow in quadrature decoder . This event is used to time updates of buffered values.

20.3.2.2 Operation

Figure 20.2. TIMER Hardware Timer/Counter Control shows the hardware timer/counter control. Software can start or stop the counter by setting the START or STOP bits in TIMERn_CMD. The counter value (CNT in TIMERn_CNT) can always be written by software to any 32-bit value.

It is also possible to control the counter through either an external pin or PRS input. This is done through the input logic for the compare/capture Channel 0. The timer/counter allows individual actions (start, stop, reload) to be taken for rising and falling input edges. This is configured in the RISEA and FALLA fields in TIMERn_CTRL. The reload value is 0 in up-count and up/down-count mode and TOP in down-count mode.

The RUNNING bit in TIMERn_STATUS indicates if the timer is running or not. If the SYNC bit in TIMERn_CFG is set, the timer is started/stopped/reloaded (external pin or PRS) when any of the other timers are started/stopped/reloaded.

The DIR bit in TIMERn_STATUS indicates the counting direction of the timer at any given time. The counter value can be read or written by software through the CNT field in TIMERn_CNT. In Up/Down-Count mode the count direction will be set to up if the CNT value is written by software.

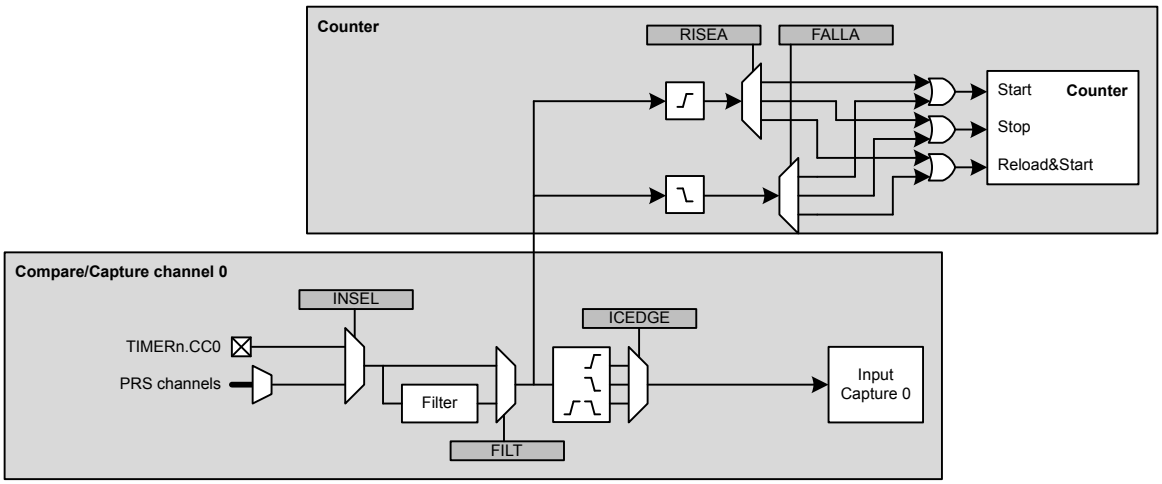


Figure 20.2. TIMER Hardware Timer/Counter Control

20.3.2.3 Clock Source

The counter can be clocked from several sources, which are all synchronized with the incoming peripheral clock for the timer. See Figure 20.3. TIMER Clock Selection.

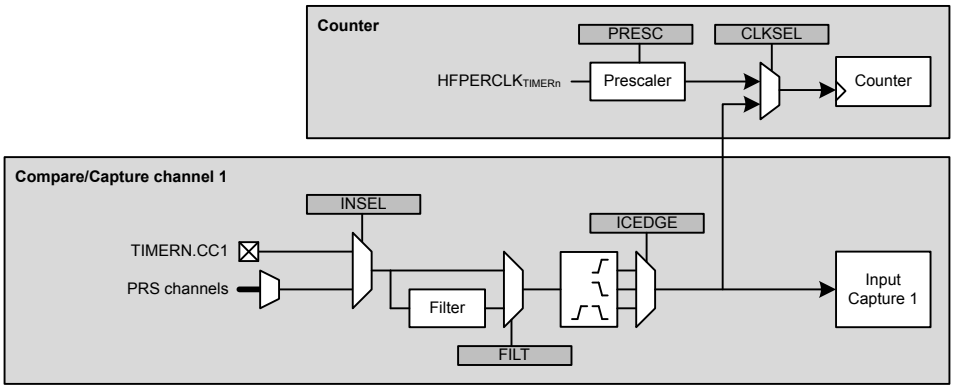


Figure 20.3. TIMER Clock Selection

20.3.2.4 Peripheral Clock

The peripheral clock for the timer ($HFPERCLK_{TIMERn}$) clocks the logic for the timer block, even when it is not the selected clock source.

All TIMER instances in this device family use EM01GRPACLK selected in CMU_EM01GRPACLKCTRL.CLKSEL as their peripheral clock source ($HFPERCLK_{TIMERn}$).

The peripheral clock to each timer can be used as a source with a configurable 10-bit prescaler. The PRESC bitfield in TIMERn_CFG sets the prescaler value, and the incoming peripheral clock will be divided by a factor of (PRESC+1). However, if 2x count mode is enabled and the compare/capture channels are configured for PWM mode, the CC output is updated on both clock edges, so prescaling the peripheral clock will produce an incorrect result. The internal prescale counter is stopped and reset when the timer is stopped.

20.3.2.5 Compare/Capture Channel 1 Input

The timer can also be clocked by positive and/or negative edges on the compare/capture channel 1 input. This input can either come from the TIMERn.CC1 pin or one of the PRS channels. The input signal must not have a higher frequency than $f_{HFPERCLK_TIMERn}/3$ when running from a pin input or a PRS input with FILT enabled in TIMERn_CCx_CFG. When running from PRS without FILT, the frequency can be as high as $f_{HFPERCLK_TIMERn}$. Note that when clocking the timer from the same pulse that triggers a start (through RISEA/FALLA in TIMERn_CTRL), the starting pulse will not update the counter value.

20.3.2.6 Underflow/Overflow From Neighboring Timer

All timers are linked together (see Figure 20.4. TIMER Connections), allowing timers to count on overflow/underflow from the lower numbered neighbouring timers to form a larger timer. Note that all timers must be set to count the same direction and less significant timer(s) can only be set to count up or down.

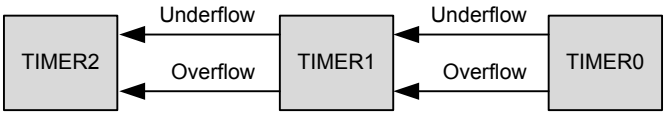


Figure 20.4. TIMER Connections

20.3.2.7 One-Shot Mode

By default, the counter counts continuously until it is stopped. If the OSMEN bit is set in the TIMERn_CFG register, however, the counter is disabled by hardware on the first *update event* (see 20.3.2.1 Events). Note that when the counter is running with CC1 as clock source and OSMEN is set, a CC1 capture event will not take place on the *update event* (CC1 rising edge) that stops the timer.

20.3.2.8 Top Value Buffer

The TIMERn_TOP register can be altered either by writing it directly or by writing to the TIMER_TOPB (buffer) register. When writing to the buffer register the TIMERn_TOPB register will be written to TIMERn_TOP on the next *update event*. Buffering ensures that the TOP value is not set below the actual count value. The TOPBV flag in TIMERn_STATUS indicates whether the TIMERn_TOPB register contains data that has not yet been written to the TIMERn_TOP register (see Figure 20.5. TIMER TOP Value Update Functionality).

Note: When writing to TIMERn_TOP register directly, the TIMERn_TOPB register value will be invalidated and the TOPBV flag will be cleared. This prevents TIMERn_TOP register from being immediately updated by an existing valid TIMERn_TOPB value during the next *update event*.

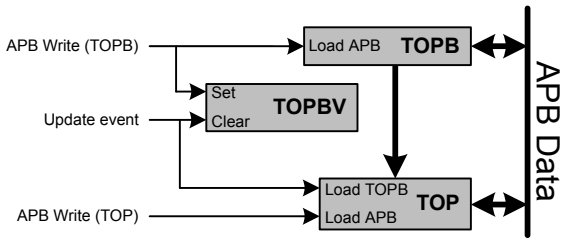


Figure 20.5. TIMER TOP Value Update Functionality

20.3.2.9 Quadrature Decoder

Quadrature decoding mode is used to track motion and determine both rotation direction and position. The quadrature decoder uses two input channels that are 90 degrees out of phase (see Figure 20.6. TIMER Quadrature Encoded Inputs).

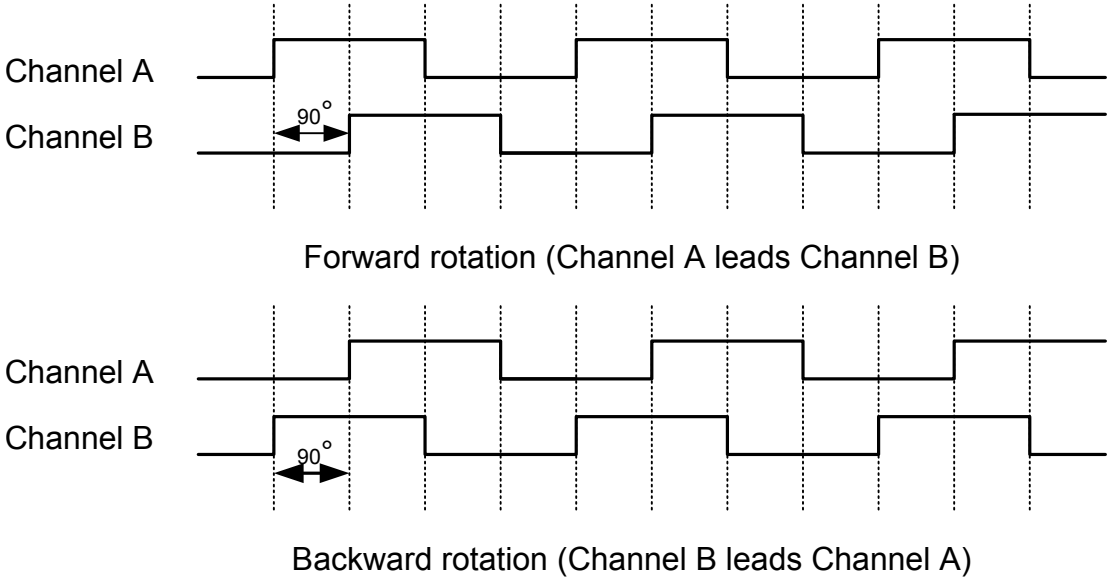


Figure 20.6. TIMER Quadrature Encoded Inputs

In the timer these inputs are tapped from the compare/capture channel 0 (Channel A) and 1 (Channel B) inputs before edge detection. The timer/counter then increments or decrements the counter, based on the phase relation between the two inputs. The DIRCHG flag in TIMERn_IF is set if the count direction changes in quadrature decoder mode. The quadrature decoder supports two channels, but if a third channel (Z-terminal) is available, this can be connected to an external interrupt and trigger a counter reset from the interrupt service routine. By connecting a periodic signal from another timer as input capture on compare/capture Channel 2, it is also possible to calculate speed and acceleration.

Note: In quadrature decoder mode, overflow and underflow triggers an *update event.*

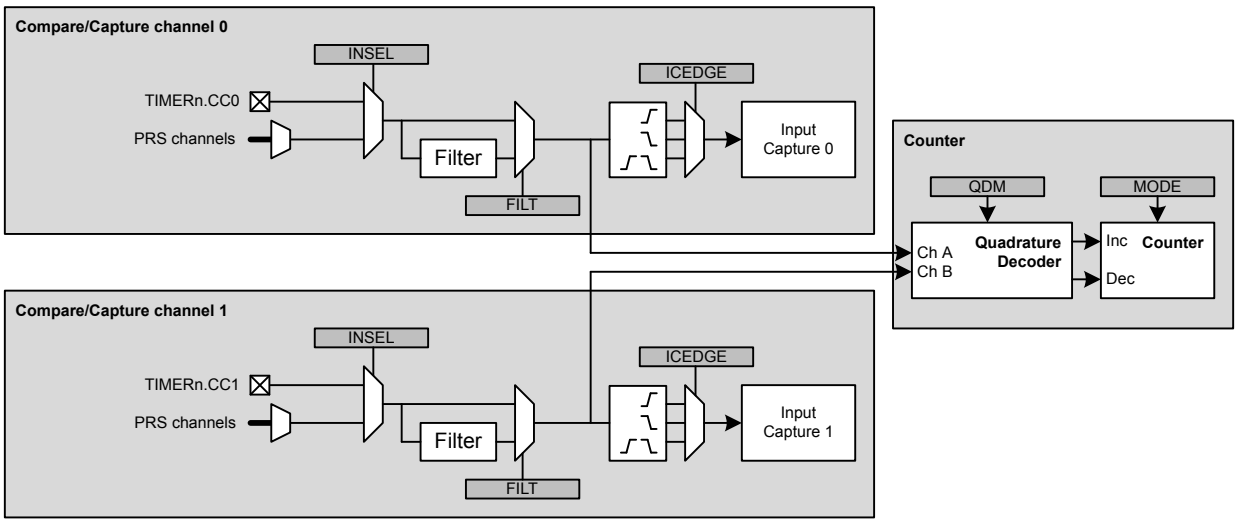


Figure 20.7. TIMER Quadrature Decoder Configuration

The quadrature decoder can be set in either X2 or X4 mode, which is configured in the QDM bit in TIMERn_CFG. See Figure 20.7. TIMER Quadrature Decoder Configuration

20.3.2.10 X2 Decoding Mode

In X2 Decoding mode, the counter increments or decrements on every edge of Channel A, see Table 20.1. TIMER Counter Response in X2 Decoding Mode and Figure 20.8. TIMER X2 Decoding Mode.

Table 20.1. TIMER Counter Response in X2 Decoding Mode

Channel B	Channel A	
	Rising	Falling
0	Increment	Decrement
1	Decrement	Increment

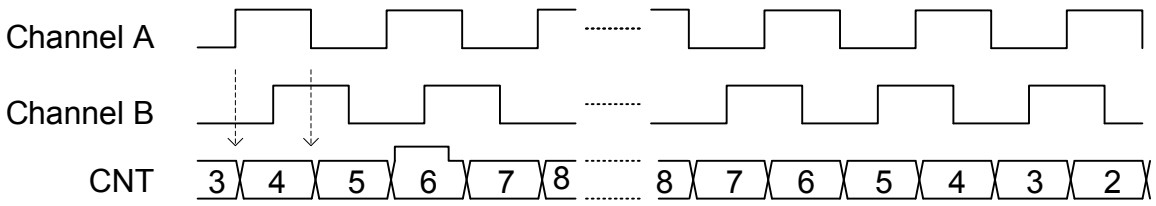


Figure 20.8. TIMER X2 Decoding Mode

20.3.2.11 X4 Decoding Mode

In X4 Decoding mode, the counter increments or decrements on every edge of Channel A and Channel B, see Figure 20.9. TIMER X4 Decoding Mode and Table 20.2. TIMER Counter Response in X4 Decoding Mode.

Table 20.2. TIMER Counter Response in X4 Decoding Mode

Opposite Channel	Channel A		Channel B	
	Rising	Falling	Rising	Falling
Channel A = 0			Decrement	Increment
Channel A = 1			Increment	Decrement
Channel B = 0	Increment	Decrement		
Channel B = 1	Decrement	Increment		

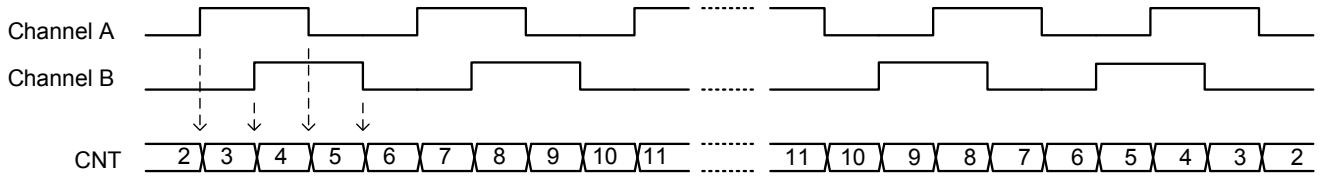


Figure 20.9. TIMER X4 Decoding Mode

20.3.2.12 Rotational Position

To calculate a position Figure 20.10. TIMER Rotational Position Equation can be used.

$$pos° = (CNT/X \times N) \times 360°$$

Figure 20.10. TIMER Rotational Position Equation

where X = Encoding type and N = Number of pulses per revolution.

20.3.3 Compare/Capture Channels

The timer contains compare/capture channels, which can be independently configured in the following modes:

1. Input Capture
2. Output Compare
3. PWM

20.3.3.1 Input Pin Logic

Each compare/capture channel can be configured as an input source for the Capture Unit or as external clock source for the timer (see Figure 20.11. TIMER Input Pin Logic). Compare/capture channels 0 and 1 are the inputs for the quadrature decoder . The input channel can be filtered before it is used, which requires the input to remain stable for up to 5 cycles in a row before the input is propagated to the output.

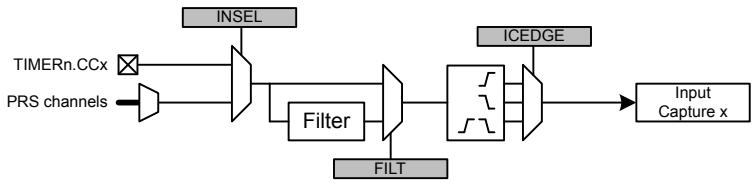


Figure 20.11. TIMER Input Pin Logic

The capture input to the timer may be selected from the dedicated CCx signal for the channel, or a PRS signal. INSEL in TIMERn_CCx_CFG determines the input to the channel. When set to PIN, the selected CCx pin will be used. When INSEL is set to PRSSYNC, a synchronous PRS channel is selected as the source. The synchronous PRS channel is determined by the SPRSSEL field in the PRS_CONSUMER_TIMERn_CCx register. Setting INSEL to PRSASYNCLEVEL or PRSASYNCPULSE selects an asynchronous PRS channel as the source. The aynchronous PRS channel is determined by the PRSSEL field in the PRS_CONSUMER_TIMERn_CCx register.

The PIN and PRSASYNCLEVEL selections are qualified by a 2-clock input sampler. To recognize and capture the incoming signal, it must be at the new level for at least 2 $HFPERCLK_{TIMERn}$ clock cycles. An additional 5 $HFPERCLK_{TIMERn}$ cycles of filtering can be applied to the signal by enabling the FILT bit in TIMERn_CCx_CFG.

The PRSASYNCPULSE selection can be used to capture higher-speed pulses on an asynchronous PRS input. The input logic for this selection does not qualify the level of the incoming signal. Instead, it will recognize positive or negative edges directly. While the pulse time can be shorter than 1 $HFPERCLK_{TIMERn}$, this mode requires at least 3 $HFPERCLK_{TIMERn}$ clocks between adjacent events. The FILT option is not used in this mode.

Synchronous PRS signals are inherently synchronized to the module clock, and the 2-clock input sampler is not used. However, it is possible to use FILT to enable the 5 $HFPERCLK_{TIMERn}$ filter when using the PRSSYNC option.

20.3.3.2 Compare/Capture Registers

The compare/capture channel registers are prefixed with TIMERn_CCx_, where the x stands for the channel number. Since the compare/capture channels serve three functions (input capture, compare, PWM), different registers are used, depending on the mode the channel is set in.

20.3.3.3 Buffered Output Update Mode

The TIMER block supports synchronous buffered updates of the output control registers used in up-count PWM output mode. This includes CCx_OC, CCx_PHASE, and CCx_DITHER. This functionality ensures glitch free updates to the PWM outputs. An overview of the buffered update hardware is shown in Figure 20.12. TIMER Buffered Output Register Function Detail.

When CFG.UPDATEMODE is set to ALL, any value that has been written to any buffer register (CCx_OCB, CCx_PHASEB, and CCx_DITHERB) since the last update will be applied to the associated direct register (CCx_OC, CCx_PHASE, and CCx_DITHER) on the next *update event*. All of the buffered output registers for a channel have a buffer valid (BV) bit in the STATUS or STATUS2 register to indicate when a new buffered value has been updated but not yet written to the associated direct register. For example, writing to CC0_OCB will set the OCBV[0] bit to '1', writing to CC1_PHASEB will set PHASEBV[1] to '1', and writing CC2_DITHERB will set DITHERBV[2] to '1'. BV bits are cleared when the buffered value has been applied to the direct output.

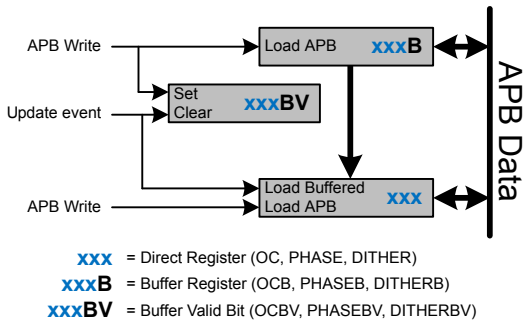


Figure 20.12. TIMER Buffered Output Register Function Detail

When CFG.UPDATEMODE is set to CMD, values written to buffered registers will not be applied to the direct registers until after CMD.UPDATECMD has been written to '1' by software. When an UPDATECMD command has been issued, the STATUS2.UPDATEBUSY bit will be set to '1'. The direct registers will then be updated on the next *update event* that occurs, and STATUS2.UPDATEBUSY will return to '0'. Figure 20.13. TIMER Buffered Output Timing with UPDATEMODE = CMD shows this relationship.

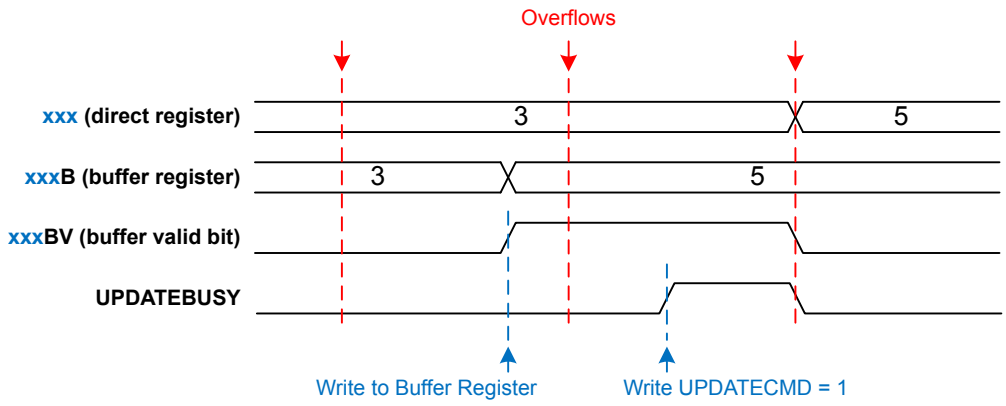


Figure 20.13. TIMER Buffered Output Timing with UPDATEMODE = CMD

20.3.3.4 Input Capture

In input capture, the counter value (TIMERn_CNT) can be captured in the Input Capture Register (TIMERn_CCx_ICF) (see Figure 20.14. TIMER Input Capture). The CCPOL bits in TIMERn_STATUS indicate the polarity of the edge that triggered the capture in TIMERn_CCx_ICF.

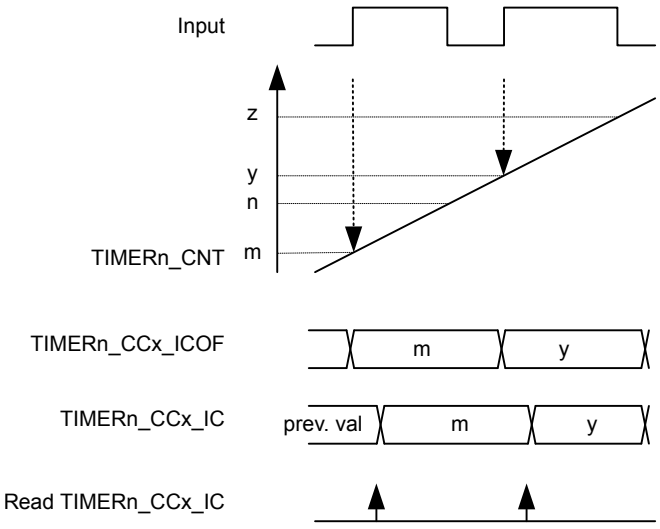


Figure 20.14. TIMER Input Capture

Input captures are buffered into a 2-entry FIFO, allowing 2 subsequent capture events to take place before a read-out is required. Reading TIMERn_CCx_ICF from software or DMA pops the oldest unread value from the FIFO. If TIMERn_CCx_ICF is read when the FIFO is empty (ICFEMPTY in TIMERn_STATUS = 1), the FIFO underflow flag for the channel (ICFUF in TIMERn_IF) will be set. The Input Capture Overflow Register (TIMERn_CCx_ICOF) always contains the newest value in the FIFO. If a new capture is triggered while the FIFO is full, the value in TIMERn_CCx_ICOF will be over-written with the latest value and the FIFO overflow flag (ICFOF in TIMERn_IF) for the channel will be set. Reading TIMERn_CCx_ICOF does not alter the FIFO contents.

The input capture FIFO also has a programmable watermark level that can be configured to generate interrupts or trigger DMA requests when a certain number of empty spots are left in the FIFO. The ICFWLFULL flag inTIMERn_IF will be set when the number of empty spots left in the FIFO is less than or equal to the watermark level programmed in TIMERn_CCx_CFG.ICFWL. At a minimum, a TIMER module will have two FIFO entries, but may have more on future devices.

The ICFEMPTY flag in TIMERn_STATUS indicates when the capture buffer is empty. When this bit reads '0', there is a valid unread capture in the FIFO.

Note: In input capture mode, the timer will only trigger interrupts when it is running.

20.3.3.5 Period/Pulse-Width Capture

Period and/or pulse-width capture can only be possible with Channel 0 (CC0), because this is the only channel that can start and stop the timer. This can be done by setting the RISEA field in TIMERn_CTRL to Clear&Start, and selecting the desired input from either external pin or PRS, see Figure 20.15. TIMER Period and/or Pulse Width Capture. For period capture, the compare/capture channel should then be set to input capture on a rising edge of the same input signal. To capture the width of a high pulse, the compare/capture channel should be set to capture on a falling edge of the input signal. To measure the low pulse-width of a signal, opposite polarities should be chosen.

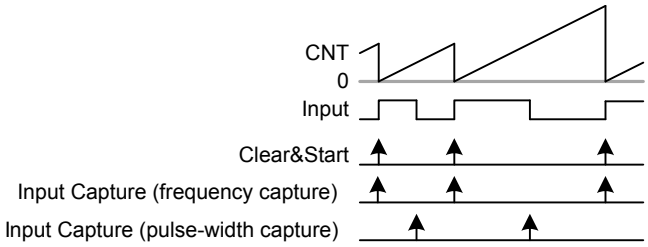


Figure 20.15. TIMER Period and/or Pulse Width Capture

20.3.3.6 Compare

Each compare/capture channel contains a comparator which outputs a compare match if the contents of TIMERn_CCx_OC matches the counter value, see Figure 20.16. TIMER Block Diagram Showing Comparison Functionality. In compare mode, each compare channel can be configured to either set, clear or toggle the output on an event (compare match, overflow or underflow). The output from each channel is represented as an alternative function on the port it is connected to, which needs to be enabled for the CC outputs to propagate to the pins.

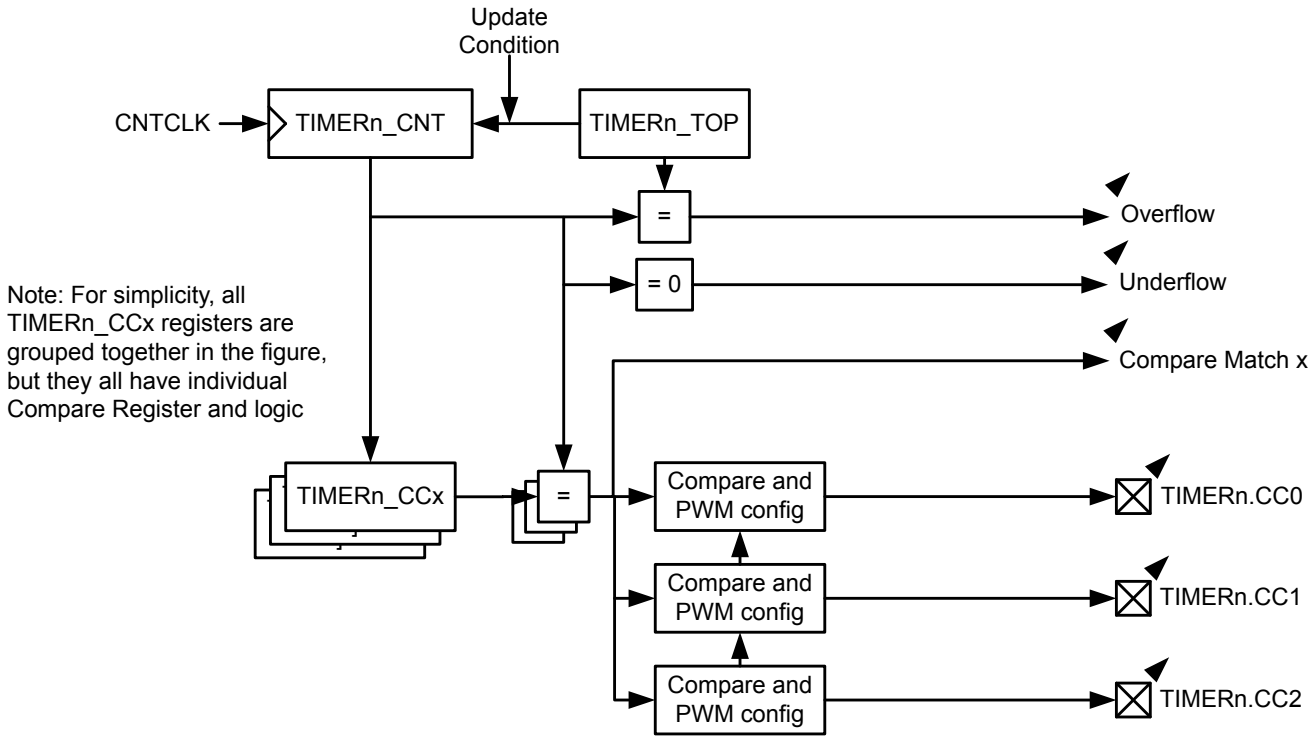


Figure 20.16. TIMER Block Diagram Showing Comparison Functionality

The compare output is delayed by one cycle to allow for full 0% to 100% PWM generation. If occurring in the same cycle, match action will have priority over overflow or underflow action.

The input selected (through PRSSEL in PRS_CONSUMER_TIMERn_CCx, INSEL and FILT in TIMERn_CCx_CFG) for the CC channel will also be sampled on compare match and the result is found in the CCPOL bits in TIMERn_STATUS. It is also possible to configure the CCPOL to always track the inputs by setting ATI in TIMERn_CFG.

Note: When using synchronous PRS sources, it is recommended to configure the PRS consumer registers prior to selecting PRS triggering to avoid any false triggers.

The COIST bit in TIMERn_CCx_CFG is the initial state of the compare/PWM output. The COIST bit can also be used as an initial value to the compare outputs on a reload-start when RSSCOIST is set in TIMERn_CFG. Also the resulting output can be inverted by setting OUTINV in TIMERn_CCx_CTRL. It is recommended to turn off the CC channel before configuring the output state to avoid any unwanted pulses on the output. The CC channel can be turned off by setting MODE to OFF in TIMER_CCx_CFG. The following figure shows the output logic for the TIMER module.

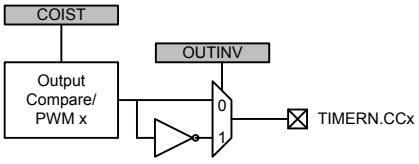


Figure 20.17. TIMER Output Logic

20.3.3.7 Compare Mode Registers

When running in output compare or PWM mode, the value in TIMERn_CCx_OC will be compared against the count value. In Compare mode the output can be configured to toggle, clear or set on compare match, overflow, and underflow through the CMOA, COFOA and CUFOA fields in TIMERn_CCx_CTRL.

TIMERn_CCx_OC can be accessed directly or through the buffer register TIMERn_CCx_OCB. When writing to the buffer register, the value in TIMERn_CCx_OCB will be written to TIMERn_CCx_OC on an *update event*. See 20.3.3.3 Buffered Output Update Mode for additional information on buffered updates. Note that when writing 0 to TIMERn_CCx_OCB in up-down count mode the OC value is updated when the timer counts from 0 to 1. Thus, the compare match for the next period will not happen until the timer reaches 0 again on the way down.

20.3.3.8 Frequency Generation (FRG)

Frequency generation (see Figure 20.18. TIMER Up-count Frequency Generation) can be achieved in compare mode by:

- Setting the counter in up-count mode
- Enabling buffering of the TOP value.
- Setting the CC channels overflow action to toggle

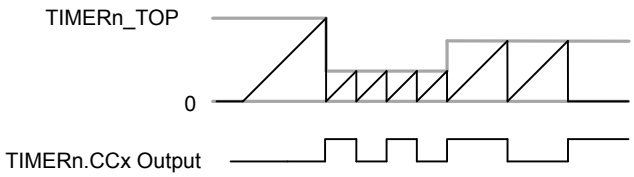


Figure 20.18. TIMER Up-count Frequency Generation

The output frequency is given by Figure 20.19. TIMER Up-count Frequency Generation Equation

$$f_{FRG} = f_{HFPERCLK_TIMERn} / [2 \times (PRESC + 1) \times (TOP + 1)]$$

Figure 20.19. TIMER Up-count Frequency Generation Equation

The figure below provides cycle accurate timing and event generation information for frequency generation.

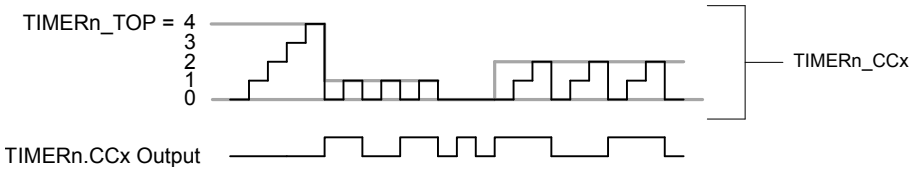


Figure 20.20. TIMER Up-count Frequency Generation Detail

20.3.3.9 Pulse-Width Modulation (PWM)

In PWM mode, TIMERn_CCx_OC is buffered to avoid glitches in the output. The settings in the Compare Output Action configuration bits are ignored in PWM mode and PWM generation is only supported for up-count and up/down-count mode.

20.3.3.10 Up-count (Single-slope) PWM

If the counter is set to up-count and the compare/capture channel is put in PWM mode, single slope PWM output will be generated (see Figure 20.21. TIMER Up-count PWM Generation). For standard edge-aligned operation, the TIMERn_CCx_PHASE register should remain at its default value of 0 (see 20.3.3.11 Phase Adjustment (Up-count PWM) for more information about phase adjustment). In up-count mode the PWM period is TOP+1 cycles and the PWM output will be high for a number of cycles equal to TIMERn_CCx_OC. This means that a constant high output is achieved by setting TIMERn_CCx_OC to TOP+1 or higher. The PWM resolution (in bits) is then given by Figure 20.22. TIMER Up-count PWM Resolution Equation.

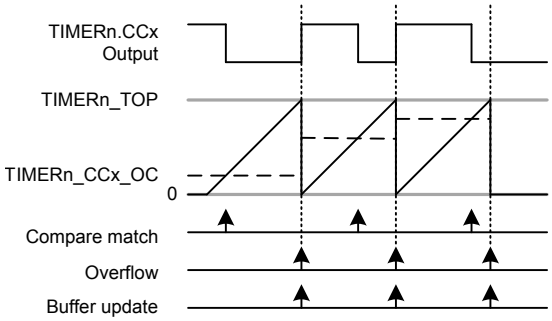


Figure 20.21. TIMER Up-count PWM Generation

$$R_{PWM_{up}} = \log(TOP+1)/\log(2)$$

Figure 20.22. TIMER Up-count PWM Resolution Equation

The PWM frequency is given by Figure 20.23. TIMER Up-count PWM Frequency Equation:

$$f_{PWM_{up}} = f_{HFPERCLK_TIMERn} / [(PRESC + 1) \times (TOP + 1)]$$

Figure 20.23. TIMER Up-count PWM Frequency Equation

The high duty cycle is given by Figure 20.24. TIMER Up-count Duty Cycle Equation (PHASE = 0)

$$DS_{up} = OCx/(TOP+1)$$

Figure 20.24. TIMER Up-count Duty Cycle Equation (PHASE = 0)

The figure below provides cycle accurate timing and event generation information for up-count mode.

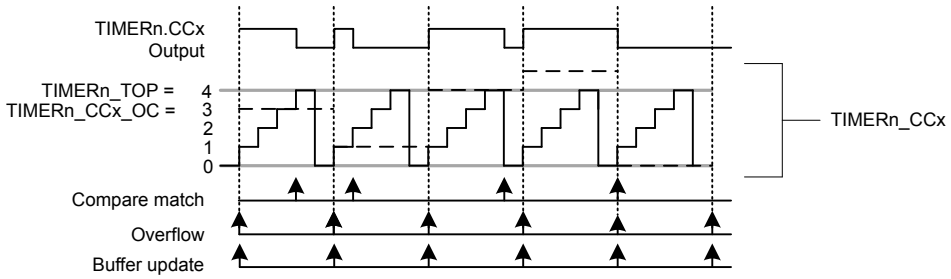


Figure 20.25. TIMER Up-count PWM Generation Detail

20.3.3.11 Phase Adjustment (Up-count PWM)

When operating in up-count PWM mode, the TIMER features programmable per-channel phase adjustment. By default the TIMERn_CCx_PHASE register for all channels is 0, and the rising edges of all channels will be aligned wth the counter overflow event. The TIMERn_CCx_PHASE register is used to offset the rising edge of each channel and produce waveforms that are not edge-aligned.

Note that TIMERn_CCx_OC still defines the falling edge for the channel when TIMERn_CCx_PHASE is adjusted, so the width (high time) of the generated pulse and the resulting duty cycle depend on TIMERn_CCx_PHASE, TIMERn_CCx_OC, and TIMERn_TOP.

When OCx > PHASEx: DS_{up} = (OCx - PHASEx)/(TOP+1)

When OCx < PHASEx: DS_{up} = (OCx + TOP+1 - PHASEx)/(TOP+1)

When OCx = PHASEx: DS_{up} = 0

Figure 20.26. TIMER Up-count Duty Cycle Equations (PHASE >= 0)

Note: The equations in Figure 20.26. TIMER Up-count Duty Cycle Equations (PHASE >= 0) are true only when PHASEx and OCx are between 0 0and TOP+1. If PHASEx >= TOP+1 and OCx < TOP+1, the duty cycle will be 0%. If OCx >= TOP+1 and PHASEx < TOP+1, the duty cycle will be 100%.

An example of three CC outputs using different PHASE settings is shown in Figure 20.27. Phase Adjustment Example. All three waveforms are based on a 7-cycle period established by the TIMERn_TOP register. CC0 and CC1 are configured with their OC registers set to values higher than their PHASE register, and begin each period in a logic low state from the previous OC match. For CC2, the PHASE register is set to a value greater than OC, which results in a high pulse that crosses the overflow boundary.

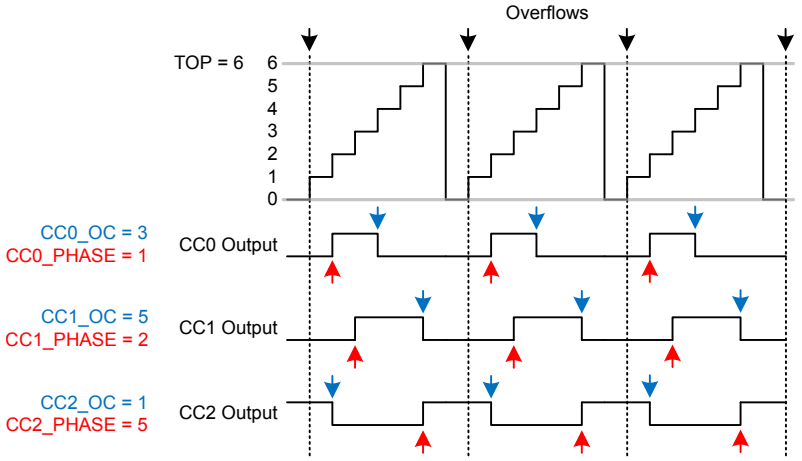


Figure 20.27. Phase Adjustment Example

TIMERn_CCx_PHASE can be accessed directly or through the buffer register TIMERn_CCx_PHASEB. When writing to the buffer register, the value in TIMERn_CCx_PHASEB will be written to TIMERn_CCx_PHASE on an *update event*. See 20.3.3.3 Buffered Output Update Mode for additional information on buffered updates.

20.3.3.12 Dither (Up-count PWM)

When operating in up-count PWM mode the TIMER features programmable dithering for the output compare edge. The dither pattern is applied over 16 PWM cycles, and extends the output compare edge by one timer tick during a programmable number of output cycles.

The TIMERn_CCx_DITHER register controls the amount of dither applied to a channel. By default the TIMERn_CCx_DITHER register for all channels is 0, and no dithering is applied (all falling edges align with TIMERn_CCx_OC). When TIMERn_CCx_DITHER is set to a non-zero value N, then N out of every 16 PWM pulses will add one cycle to the pulse width, such that the falling edge occurs at TIMERn_CCx_OC + 1.

A fractional divider determines which PWM cycles contain wider pulses. This ensures that the wider pulses will be distributed as evenly as possible, and in a predictable pattern. Figure 20.28. Cycle-by-Cycle Effects of DITHER Adjustment illustrates the extended PWM pulses across 16 cycles for each setting of DITHER. The highlighted cells show which cycles will have an extended pulse. For example, with DITHER = 1, the pulse will be extended during the first cycle only. With DITHER = 8, the pulse will be extended during every other cycle.

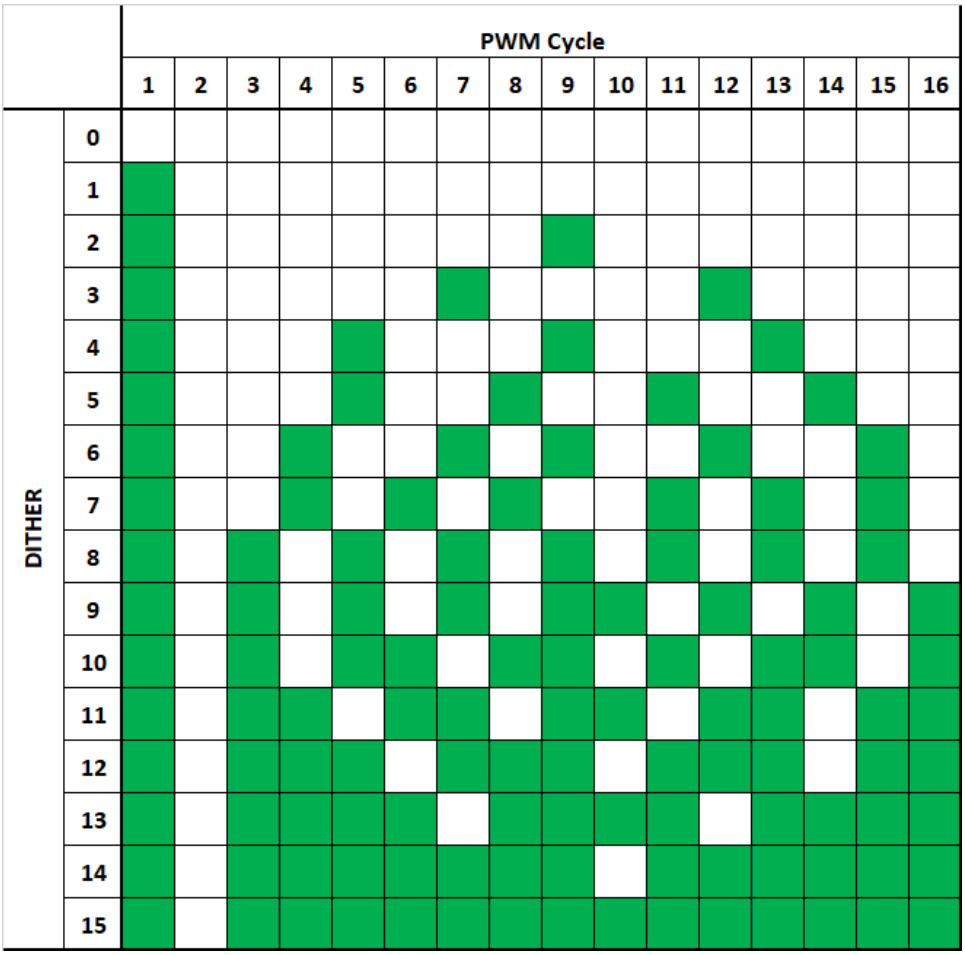


Figure 20.28. Cycle-by-Cycle Effects of DITHER Adjustment

Note: The fractional divider is reset every time DITHER is updated.

TIMERn_CCx_DITHER can be accessed directly or through the buffer register TIMERn_CCx_DITHERB. When writing to the buffer register, the value in TIMERn_CCx_DITHERB will be written to TIMERn_CCx_DITHER on an *update event*. See 20.3.3.3 Buffered Output Update Mode for additional information on buffered updates.

20.3.3.13 2x Count Mode (Up-count)

When the timer is set in 2x mode, the TIMER will count up by two for every (prescaled) clock. This will in effect make any odd Top value be rounded down to the closest even number. Similarly, any odd OC value will generate a match on the closest lower even value as shown in Figure 20.29. TIMER CC Out in 2x Mode

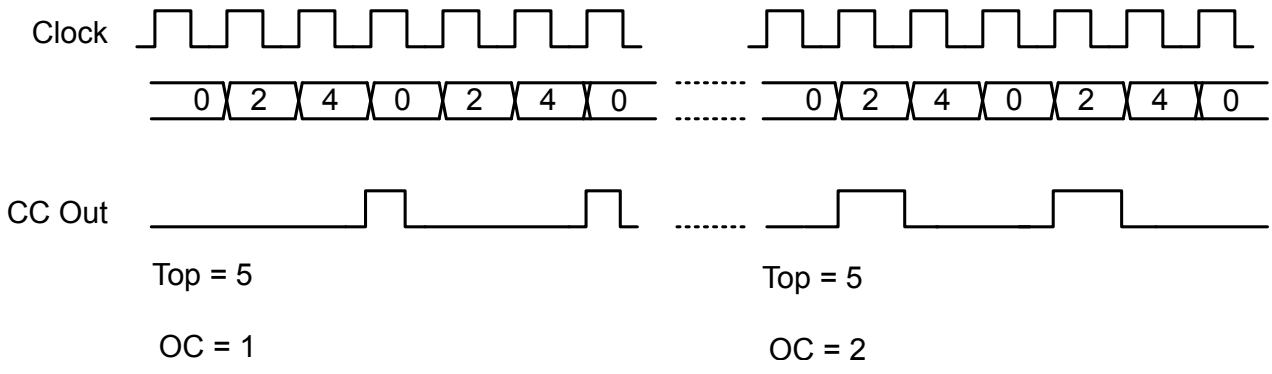


Figure 20.29. TIMER CC Out in 2x Mode

The PWM resolution is given by Figure 20.30. TIMER 2x PWM Resolution Equation.

$$R_{PWM_{2xmode}} = \log(TOP/2+1)/\log(2)$$

Figure 20.30. TIMER 2x PWM Resolution Equation

The PWM frequency is given by Figure 20.31. TIMER 2x Mode PWM Frequency Equation (Up-count):

$$f_{PWM_{2xmode}} = f_{HFPERCLK_TIMERn} / [(PRESC + 1) \times (\text{floor}(TOP/2)+1)]$$

Figure 20.31. TIMER 2x Mode PWM Frequency Equation (Up-count)

The high duty cycle is given by Figure 20.32. TIMER 2x Mode Duty Cycle Equation (PHASE = 0)

$$DS_{2xmode} = OCx/((\text{floor}(TOP/2)+1)*2)$$

Figure 20.32. TIMER 2x Mode Duty Cycle Equation (PHASE = 0)

20.3.3.14 Up/Down-count (Dual-slope) PWM

If the counter is set to up-down count and the compare/capture channel is put in PWM mode, dual slope PWM output will be generated by Figure 20.33. TIMER Up/Down-count PWM Generation. The resolution (in bits) is given by Figure 20.34. TIMER Up/Down-count PWM Resolution Equation.

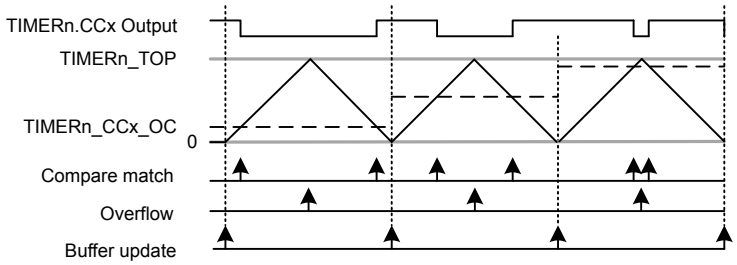


Figure 20.33. TIMER Up/Down-count PWM Generation

$$R_{PWM_{up/down}} = \log(TOP+1)/\log(2)$$

Figure 20.34. TIMER Up/Down-count PWM Resolution Equation

The PWM frequency is given by Figure 20.35. TIMER Up/Down-count PWM Frequency Equation:

$$f_{PWM_{up/down}} = f_{HFPERCLK_TIMERn}/ (2 \times (PRESC + 1) \times TOP))$$

Figure 20.35. TIMER Up/Down-count PWM Frequency Equation

The high duty cycle is given by Figure 20.36. TIMER Up/Down-count Duty Cycle Equation

$$DS_{up/down} = OCx/TOP$$

Figure 20.36. TIMER Up/Down-count Duty Cycle Equation

The figure below provides cycle accurate timing and event generation information for up-count mode.

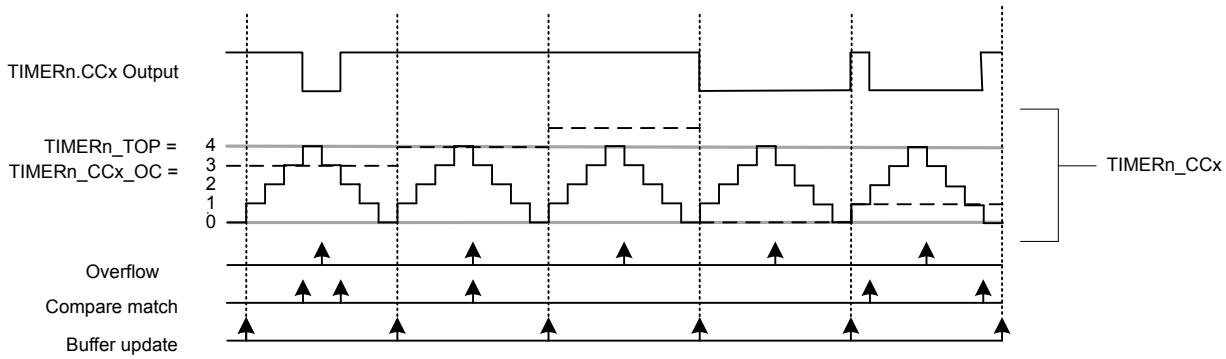


Figure 20.37. TIMER Up/Down-count PWM Generation

20.3.3.15 2x Count Mode (Up/Down-count)

When the timer is set in 2x mode, the TIMER will count up/down by two. This will in effect make any odd Top value be rounded down to the closest even number. Similarly, any odd OC value will generate a match on the closest lower even value as shown in Figure 20.38. TIMER CC Out in 2x Mode

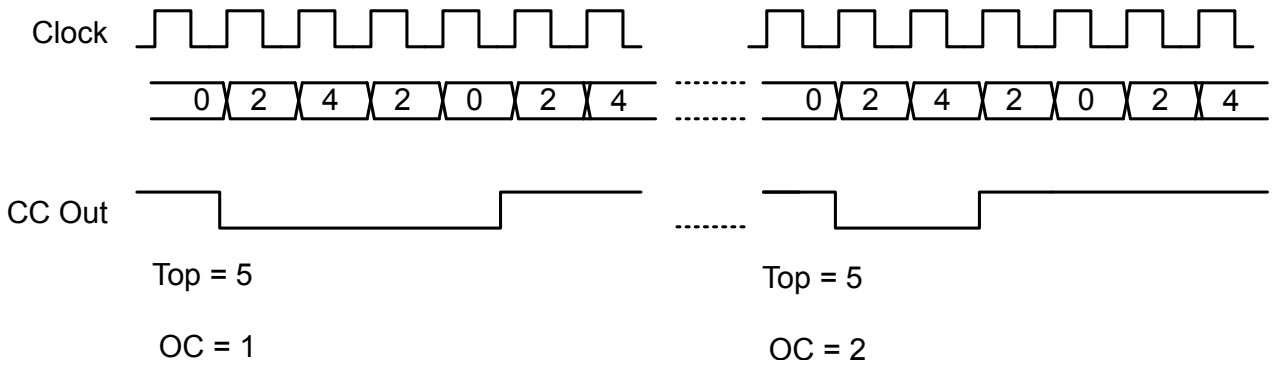


Figure 20.38. TIMER CC Out in 2x Mode

Figure 20.39. TIMER 2x PWM Resolution Equation.

$$R_{PWM_{2xmode}} = \log(TOP/2+1)/\log(2)$$

Figure 20.39. TIMER 2x PWM Resolution Equation

The PWM frequency is given by Figure 20.40. TIMER 2x Mode PWM Frequency Equation (Up/Down-count):

$$f_{PWM_{2xmode}} = f_{HFPERCLK_TIMERn} / (2 \times (PRESC + 1) \times (\text{floor}(TOP/2)))$$

Figure 20.40. TIMER 2x Mode PWM Frequency Equation (Up/Down-count)

The high duty cycle is given by two equations based on the OCx values.Figure 20.41. TIMER 2x Mode Duty Cycle Equation for OCx = 1 or OCx = Even and Figure 20.42. TIMER 2x Mode Duty Cycle Equation for all Other OCx = Odd Values

$$DS_{2xmode} = (OCx*2)/(\text{floor}(TOP/2)*4)$$

Figure 20.41. TIMER 2x Mode Duty Cycle Equation for OCx = 1 or OCx = Even

$$DS_{2xmode} = (OCx*2 - OCx)/(\text{floor}(TOP/2)*4)$$

Figure 20.42. TIMER 2x Mode Duty Cycle Equation for all Other OCx = Odd Values

20.3.3.16 Re-Timing PWM Outputs

PWM outputs are normally synchrous to the TIMER peripheral clock. However for radio applications, it can be desirable to synchronize PWM edges to radio clocks to reduce the interference with RF signalling.

Re-timing is enabled by setting the RETIMEEN bit in TIMERn_CFG to 1. When RETIMEEN is enabled, PWM X2CNT mode should not be enabled. Doing so may result in unpredictable PWM behavior.

Direct re-timing is supported at peripheral clock frequencies up to 50 MHz. For higher peripheral clock frequencies, set the RETIMESEL bit in TIMERn_CFG to 1. This allows PWM outputs to be re-timed at frequencies up to 80 MHz, but will introduce up to 1 $HFPERCLK_{TIMERn}$ cycle of jitter between the PWM outputs.

20.3.3.17 Timer Configuration Lock

To prevent software errors from making changes to the timer configuration, a configuration lock is available. Writing any value but 0xCE80 to LOCKKEY in TIMERn_LOCK will lock writes to TIMERn_CTRL, TIMERn_CFG, TIMERn_CMD, TIMERn_TOP, TIMERn_TOPB, TIMERn_CNT, TIMERn_CCx_CTRL, TIMERn_CCx_CFG, TIMERn_CCx_OC, and TIMERn_CCx_OCB. To unlock the registers, write 0xCE80 to LOCKKEY in TIMERn_LOCK. The value of TIMERLOCKSTATUS in TIMERn_STATUS is 1 when the lock is active, and 0 when the registers are unlocked.

20.3.4 Dead-Time Insertion Unit

Some timer modules include a Dead-Time Insertion unit suitable for motor control applications. Refer to the device data sheet to check which timer instances have this feature. The example settings in this section are for TIMER0, but identical settings can be used for other timers with DTI as well. The Dead-Time Insertion Unit aims to make control of brushless DC (BLDC) motors safer and more efficient by introducing complementary PWM outputs with dead-time insertion and fault handling, see Figure 20.43. TIMER Dead-Time Insertion Unit Overview.

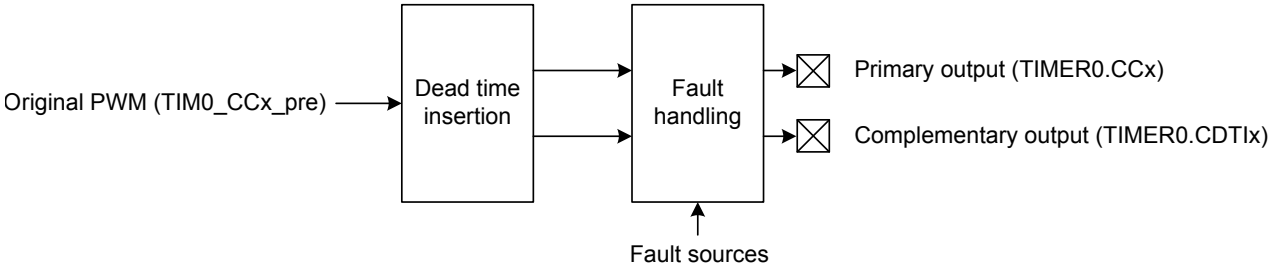


Figure 20.43. TIMER Dead-Time Insertion Unit Overview

When used for motor control, the PWM outputs TIMERn.CC0, TIMERn.CC1 and TIMERn.CC2 are often connected to the high-side transistors of a triple half-bridge setup (UH, VH and WH), and the complementary outputs connected to the respective low-side transistors (UL, VL, WL shown in Figure 20.44. TIMER Triple Half-Bridge). Transistors used in such a bridge often do not open/close instantaneously, and using the exact complementary inputs for the high and low side of a half-bridge may result in situations where both gates are open. This can give unnecessary current-draw and short circuit the power supply. The DTI unit provides dead-time insertion to deal with this problem.

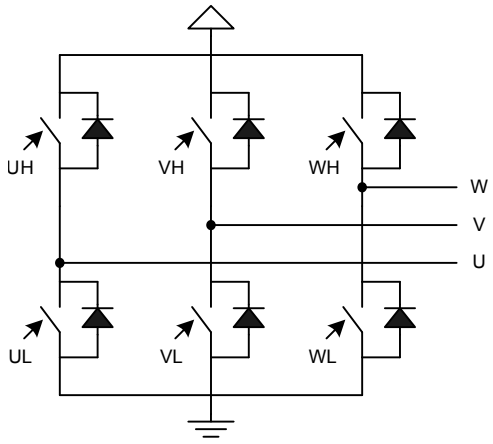


Figure 20.44. TIMER Triple Half-Bridge

For each of the 3 compare-match outputs of TIMER0, an additional complementary output is provided by the DTI unit. These outputs, named TIMERn.CDTI0, TIMERn.CDTI1 and TIMERn.CDTI2 are provided to make control of e.g. 3-channel BLDC or permanent magnet AC (PMAC) motors possible using only a single timer, see Figure 20.45. TIMER Overview of Dead-Time Insertion Block for a Single PWM Channel.

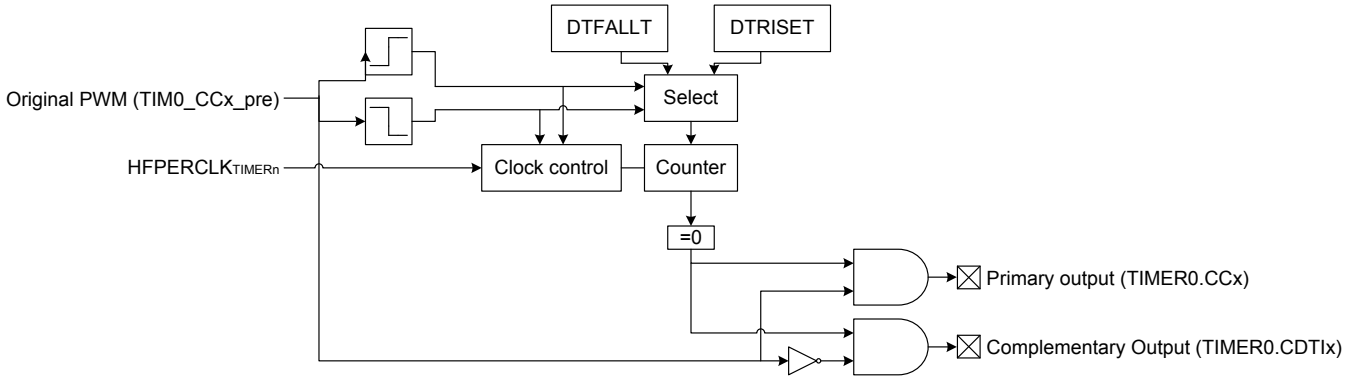


Figure 20.45. TIMER Overview of Dead-Time Insertion Block for a Single PWM Channel

The DTI unit is enabled by setting DTEN in TIMER0_DTCFG. In addition to providing the complementary outputs, the DTI unit then also overrides the compare match outputs from the timer.

The DTI unit gives the rising edges of the PWM outputs and the rising edges of the complementary PWM outputs a configurable time delay. By doing this, the DTI unit introduces a dead-time where both the primary and complementary outputs in a pair are inactive as seen in Figure 20.46. TIMER Polarity of Both Signals are Set as Active-High.

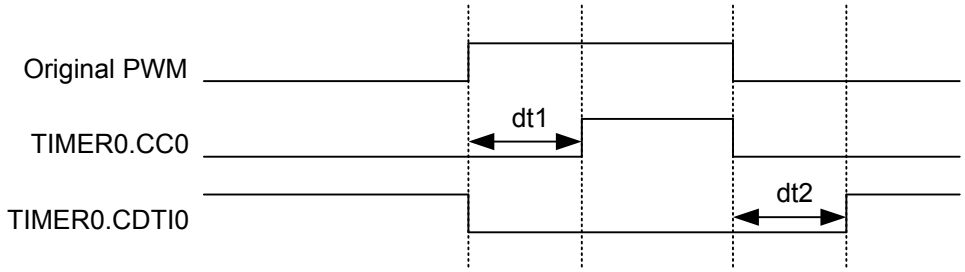


Figure 20.46. TIMER Polarity of Both Signals are Set as Active-High

Dead-time is specified individually for the rising and falling edge of the original PWM. These values are shared across all the three PWM channels of the DTI unit. A single prescaler value is provided for the DTI unit, meaning that both the rising and falling edge dead-times share prescaler value. The prescaler divides the $HFPERCLK_{TIMER0}$ by a configurable factor between 1 and 1024, which is set in the DTPRESC field in TIMER0_DTTIMECFG. The rising and falling edge dead-times are configured in DTRISET and DTFALLT in TIMER0_DTTIMECFG to any number between 1-64 $HFPERCLK_{TIMER0}$ cycles.

The DTAR and DTFATS bits in TIMER0_DTCFG control the DTI output behavior when the timer stops. By default the DTI block stops when the timer is stopped. Setting the DTAR bit will cause the DTI output on channel 0 to continue when the timer is stopped. DTAR effects only channel 0. See 20.3.4.2 PRS Channel as a Source for an example of when this can be used. While in this mode the undivided $HFPERCLK_{TIMER0}$ (DTPRESC=0) is always used regardless of the programmed DTPRESC value in TIMER0_DTTIMECFG. This means that rise and fall dead times are calculated assuming DTPRESC = 0.

When the timer stops, DTI outputs are frozen by default, preserving their last state. To allow the outputs to go to a safe state, program the DTFA field of the TIMER0_DTFCFG register to the safe values and set the DTFATS bitfield in the TIMER0_DTCFG register. Note that when DTAR is also set, DTAR has priority over DTFATS for DTI channel 0 output.

The following table shows the DTI output when the timer is halted.

Table 20.3. DTI Output When Timer Halted

DTAR	DTFATS	State
0	0	frozen
0	1	safe
1	0	running
1	1	running

20.3.4.1 Output Polarity

The value of the primary and complementary outputs in a pair will never be set active at the same time by the DTI unit. The polarity of the outputs can be changed if this is required by the application. The active values of the primary and complementary outputs are set by the DTIPOL and DTCINV bits in the TIMER0_DTCTRL register. The DTIPOL bit of this register specifies the base polarity. If DTIPOL = 0, then the outputs are active-high, and if DTIPOL = 1 they are active-low. The relative phase of the primary and complementary outputs is not changed by DTIPOL, as the polarity of both outputs is changed, see Figure 20.47. TIMER Output Polarities.

In some applications, it may be required that the primary outputs are active-high, while the complementary outputs are active-low. This can be accomplished by manipulating the DTCINV bit of the TIMER0_DTCTRL register, which inverts the polarity of the complementary outputs relative to the primary outputs. As an example, DTIPOL = 0 and DTCINV = 0 results in outputs with opposite phase and active-high states. Similarly, DTIPOL = 1 and DTCINV = 1 results in outputs with equal phase and the primary output will be active-high while the complementary will be active-low.

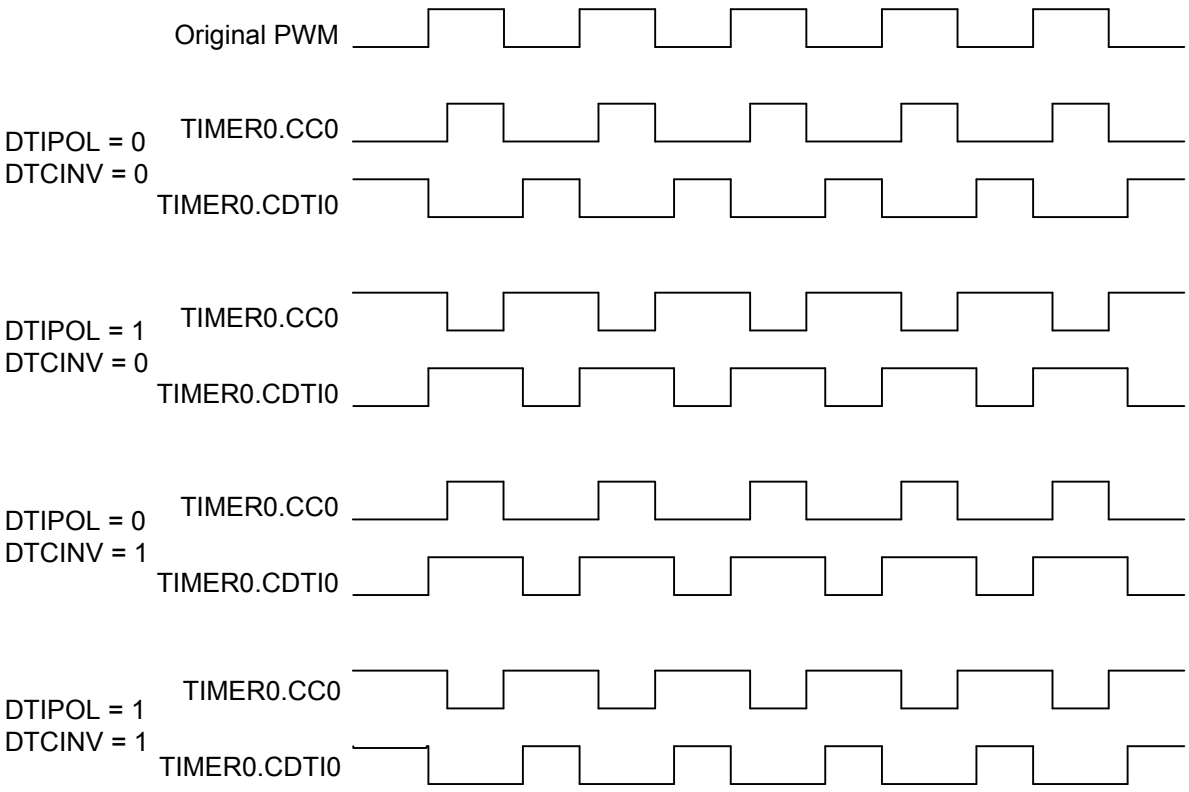


Figure 20.47. TIMER Output Polarities

Output generation on the individual DTI outputs can be disabled by configuring TIMER0_DTOGEN. When output generation on an output is disabled that output will go to and stay in its inactive state.

20.3.4.2 PRS Channel as a Source

A PRS channel can be used as input to the DTI module instead of the PWM output from the timer for DTI channel 0. Setting DTPRSEN in TIMER0_DTCFG will override the source of the first DTI channel, driving TIMER0.CC0 and TIMER0.CDTI0, with the value on the PRS channel. The rest of the DTI channels will continue to be driven by the PWM output from the timer. The input PRS channel is chosen within the PRS module with PRSSEL in the PRS_CONSUMER_TIMERn_DTI register. Note that the timer must be running even when PRS is used as the DTI source. However, if it is required to keep the DTI channel 0 running even when the timer is stopped, set DTAR in TIMER0_DTCFG. When this bit is set, it uses DTPRESC=0 regardless of the value programmed in DTPRESC in TIMER0_DTTIMECFG.

Note: When using synchronous PRS sources, it is recommended to configure the PRS consumer registers prior to selecting PRS triggering to avoid any false triggers.

The DTI prescaler, set by DTPRESC in TIMER0_DTTIMECFG determines the accuracy with which the DTI can insert dead-time into a PRS signal. The maximum dead-time error equals DTIPRESC+1 clock cycles. With DTIPRESC = 0, the inserted dead-times are therefore accurate, but they may be inaccurate for larger prescaler settings.

20.3.4.3 Fault Handling

The fault handling system of the DTI unit allows the outputs of the DTI unit to be put in a well-defined state in case of a fault. This hardware fault handling system enables a fast reaction to faults, reducing the possibility of damage to the system.

The fault sources which trigger a fault in the DTI module are determined by the bitfields of TIMER0_DTFCFG register. Any combination of the available error sources can be selected:

- PRS source 1, determined by PRSSEL in PRS_CONSUMER_TIMERn_DTIFS1
- PRS source 2, determined by PRSSEL in PRS_CONSUMER_TIMERn_DTIFS2
- Debugger
- Core Lockup

One or two PRS channels can be used as an error source. When PRS source 1 is selected as an error source, PRSSEL in PRS_CONSUMER_TIMERn_DTIFS1 determines which PRS channel is used for this source. PRSSEL in PRS_CONSUMER_TIMERn_DTIFS2 determines which PRS channel is selected as PRS source 2. Note that for Core Lockup, the LOCKUPRMODE in EMU_RSTCTRL must be set to DISABLED. Otherwise this will generate a full reset of the chip.

Note: When using synchronous PRS sources, it is recommended to configure the PRS consumer registers prior to selecting PRS triggering to avoid any false triggers.

20.3.4.4 Action on Fault

When a fault occurs, the bit representing the fault source is set in TIMER0_DTFAULT register, and the outputs from the DTI unit are set to a well-defined state. The following options are available, and can be enabled by configuring DTFA in TIMER0_DTFCFG:

- Set outputs to inactive level
- Clear outputs
- Tristate outputs

With the first option enabled, the output state in case of a fault depends on the polarity settings for the individual outputs. An output set to be active high will be set low if a fault is detected, while an output set to be active low will be driven high.

When a fault occurs, the fault source(s) can be read out from TIMER0_DTFAULT register.

Additionally a fault action can also be triggered when the timer stops if DTFATS in TIMER0_DTCFG is set. This allows the DTI output to go to safe state specified by DTFA in TIMER0_DTFCFG when the timer stops. When DTAR and DTFATS in TIMER0_DTCFG are both set, DTI channel 0 keeps running even when the timer stops. This is useful when DTI channel 0 has an input coming from PRS.

20.3.4.5 Exiting Fault State

When a fault is triggered by the PRS system, software intervention is required to re-enable the outputs of the DTI unit. This is done by manually clearing bits in the TIMER0_DTFAULT register. If the fault source as determined by checking TIMER0_DTFAULT is the debugger alone, the outputs can be automatically restarted when the debugger exits. To enable automatic restart set DTDAS in TIMER0_DTCFG. When an automatic restart occurs the DTDBGF bit in TIMER0_DTFAULT will be automatically cleared by hardware. If any other bits in the TIMER0_DTFAULT register are set when the hardware clears DTDBGF the DTI module will not exit the fault state.

20.3.4.6 DTI Configuration Lock

To prevent software errors from making changes to the DTI configuration, a configuration lock is available. Writing any value but 0xCE80 to LOCKKEY in TIMER0_DTLOCK locks writes to registers TIMER0_DTCFG, TIMER0_DTFCFG, TIMER0_DTCTRL, and TIMER0_DTTIMECFG. To unlock the registers, write 0xCE80 to LOCKKEY in TIMER0_DTLOCK. The value of DTILOCKSTATUS in TIMERn_STATUS is 1 when the lock is active, and 0 when the registers are unlocked.

20.3.5 Debug Mode

When the CPU is halted in debug mode, the timer can be configured to either continue to run or to be frozen. This is configured in DEBUGRUN in TIMERn_CFG.

20.3.6 Interrupts, DMA and PRS Output

The timer can generate several type of output events:

- Counter Underflow
- Counter Overflow
- Quadrature Decoder Direction Change
- Compare match or input capture (one per compare/capture channel)

Each of the events has its own interrupt flag. Also, there are interrupt flags for each compare/capture channel which are set on FIFO overflow or underflow in capture mode. FIFO overflow happens when a new capture over-writes an old unread capture in TIMERn_CCx_ICF. FIFO underflow happens when software reads TIMERn_CCx_ICF while the FIFO is empty.

If the interrupt flags are set and the corresponding interrupt enable bits in TIMERn_IEN are set high, the timer will send out an interrupt request. Each of the events may optionally trigger signals to PRS channels. The PRSCONF field in TIMERn_CCx_CFG determines how PRS events are generated. When PRSCONF is set to PULSE, and event will lead to a one $HFPERCLK_{TIMERn}$ cycle high pulse on individual PRS outputs. Setting PRSCONF to LEVEL will make the PRS output follow the compare match output. Interrupts are cleared by setting the corresponding bit in the TIMERn_IF_CLR register.

Each of the events will also set a DMA request when they occur. The different DMA requests are cleared when certain acknowledge conditions are met, see Table 20.4. TIMER DMA Events. Events which clear the DMA requests do not clear interrupt flags. Software must still manually clear the interrupt flag if interrupts are in use.

If DMACLRACT is set in TIMERn_CFG, the DMA request is cleared when the triggered DMA channel is active, without having to access any timer registers. This is useful in cases where a timer event is used to trigger a DMA transfer in output compare or PWM mode that does not target the OC or OCB registers. DMACLRACT is not applicable in input capture mode.

Table 20.4. TIMER DMA Events

Event	Acknowledge/Clear
Underflow/Overflow	Read or write to TIMERn_CNT or TIMERn_TOPB
CC0 Input Capture - ICFWLFULL0 flag set	ICFEMPTY0 flag set (read FIFO via TIMERn_CC0_ICF)
CC1 Input Capture - ICFWLFULL1 flag set	ICFEMPTY1 flag set (read FIFO via TIMERn_CC1_ICF)
CC2 Input Capture - ICFWLFULL2 flag set	ICFEMPTY2 flag set (read FIFO via TIMERn_CC2_ICF)
CC3 Input Capture - ICFWLFULL3 flag set	ICFEMPTY3 flag set (read FIFO via TIMERn_CC3_ICF)
CC3 Input Capture - ICFWLFULL4 flag set	ICFEMPTY4 flag set (read FIFO via TIMERn_CC4_ICF)
CC3 Input Capture - ICFWLFULL5 flag set	ICFEMPTY5 flag set (read FIFO via TIMERn_CC5_ICF)
CC3 Input Capture - ICFWLFULL6 flag set	ICFEMPTY6 flag set (read FIFO via TIMERn_CC6_ICF)
CC0 Output Compare / PWM - Match event	Write TIMERn_CC0_OC or TIMERn_CC0_OCB
CC1 Output Compare / PWM - Match event	Write TIMERn_CC1_OC or TIMERn_CC1_OCB
CC2 Output Compare / PWM - Match event	Write TIMERn_CC2_OC or TIMERn_CC2_OCB
CC3 Output Compare / PWM - Match event	Write TIMERn_CC3_OC or TIMERn_CC3_OCB
CC4 Output Compare / PWM - Match event	Write TIMERn_CC4_OC or TIMERn_CC4_OCB
CC5 Output Compare / PWM - Match event	Write TIMERn_CC5_OC or TIMERn_CC5_OCB
CC6 Output Compare / PWM - Match event	Write TIMERn_CC6_OC or TIMERn_CC6_OCB

20.3.7 GPIO Input/Output

The TIMERn.CCx inputs/outputs and TIMERn.CDTIx outputs are accessible as alternate functions through GPIO. Each pin connection can be enabled/disabled separately using the GPIO module control registers. See the device data sheet for the available locations for each signal.

20.4 TIMER Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	TIMER_IPVERSION	R	IP Version ID
0x004	TIMER_CFG	RW CONFIG	Configuration Register
0x008	TIMER_CTRL	RW SYNC	Control Register
0x00C	TIMER_CMD	W SYNC	Command Register
0x010	TIMER_STATUS	RH	Status Register
0x014	TIMER_STATUS2	RH	Status Reg for Phase and Dither
0x018	TIMER_IF	RWH INTFLAG	Interrupt Flag Register
0x01C	TIMER_IEN	RW	Interrupt Enable Register
0x020	TIMER_TOP	RWH SYNC	Counter Top Value Register
0x024	TIMER_TOPB	RW SYNC	Counter Top Value Buffer Register
0x028	TIMER_CNT	RWH SYNC	Counter Value Register
0x02C	TIMER_LOCK	W	TIMER Configuration Lock Register
0x030	TIMER_EN	RW ENABLE	Module En
0x060	TIMER_CCx_CFG	RW CONFIG	CC Channel Configuration Register
0x064	TIMER_CCx_CTRL	RW SYNC	CC Channel Control Register
0x068	TIMER_CCx_PHASE	RWH SYNC	Channel Phase Register
0x06C	TIMER_CCx_PHASEB	RW SYNC	Channel Phase Buffer Register
0x070	TIMER_CCx_OC	RWH SYNC	OC Channel Value Register
0x074	TIMER_CCx_OCB	RW SYNC	OC Channel Value Buffer Register
0x078	TIMER_CCx_DITHER	RWH SYNC	Dither Value Register
0x07C	TIMER_CCx_DITHERB	RW SYNC	Dither Value Buffer Register
0x084	TIMER_CCx_ICF	RH(r)	IC Channel Value Register
0x088	TIMER_CCx_ICOF	RH SYNC	IC Channel Value Overflow Register
0x1B0	TIMER_DTCFG	RW CONFIG	DTI Configuration Register
0x1B4	TIMER_DTTIMECFG	RW CONFIG	DTI Time Configuration Register
0x1B8	TIMER_DTFCFG	RW CONFIG	DTI Fault Configuration Register
0x1BC	TIMER_DTCTRL	RW SYNC	DTI Control Register
0x1C0	TIMER_DTOGEN	RW SYNC	DTI Output Generation Enable Register
0x1C4	TIMER_DTFAULT	RH	DTI Fault Register
0x1C8	TIMER_DTFAULTC	W SYNC	DTI Fault Clear Register
0x1CC	TIMER_DTLOCK	W	DTI Configuration Lock Register
0x1000	TIMER_IPVERSION_SET	R	IP Version ID
0x1004	TIMER_CFG_SET	RW CONFIG	Configuration Register
0x1008	TIMER_CTRL_SET	RW SYNC	Control Register
0x100C	TIMER_CMD_SET	W SYNC	Command Register
0x1010	TIMER_STATUS_SET	RH	Status Register
0x1014	TIMER_STATUS2_SET	RH	Status Reg for Phase and Dither
0x1018	TIMER_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x101C	TIMER_IEN_SET	RW	Interrupt Enable Register

Offset	Name	Type	Description
0x1020	TIMER_TOP_SET	RWH SYNC	Counter Top Value Register
0x1024	TIMER_TOPB_SET	RW SYNC	Counter Top Value Buffer Register
0x1028	TIMER_CNT_SET	RWH SYNC	Counter Value Register
0x102C	TIMER_LOCK_SET	W	TIMER Configuration Lock Register
0x1030	TIMER_EN_SET	RW ENABLE	Module En
0x1060	TIMER_CCx_CFG_SET	RW CONFIG	CC Channel Configuration Register
0x1064	TIMER_CCx_CTRL_SET	RW SYNC	CC Channel Control Register
0x1068	TIMER_CCx_PHASE_SET	RWH SYNC	Channel Phase Register
0x106C	TIMER_CCx_PHASEB_SET	RW SYNC	Channel Phase Buffer Register
0x1070	TIMER_CCx_OC_SET	RWH SYNC	OC Channel Value Register
0x1074	TIMER_CCx_OCB_SET	RW SYNC	OC Channel Value Buffer Register
0x1078	TIMER_CCx_DITHER_SET	RWH SYNC	Dither Value Register
0x107C	TIMER_CCx_DITHERB_SET	RW SYNC	Dither Value Buffer Register
0x1084	TIMER_CCx_ICF_SET	RH(r)	IC Channel Value Register
0x1088	TIMER_CCx_ICOF_SET	RH SYNC	IC Channel Value Overflow Register
0x11B0	TIMER_DTCFG_SET	RW CONFIG	DTI Configuration Register
0x11B4	TIMER_DTTIMECFG_SET	RW CONFIG	DTI Time Configuration Register
0x11B8	TIMER_DTFCFG_SET	RW CONFIG	DTI Fault Configuration Register
0x11BC	TIMER_DTCTRL_SET	RW SYNC	DTI Control Register
0x11C0	TIMER_DTOGEN_SET	RW SYNC	DTI Output Generation Enable Register
0x11C4	TIMER_DTFAULT_SET	RH	DTI Fault Register
0x11C8	TIMER_DTFAULTC_SET	W SYNC	DTI Fault Clear Register
0x11CC	TIMER_DTLOCK_SET	W	DTI Configuration Lock Register
0x2000	TIMER_IPVERSION_CLR	R	IP Version ID
0x2004	TIMER_CFG_CLR	RW CONFIG	Configuration Register
0x2008	TIMER_CTRL_CLR	RW SYNC	Control Register
0x200C	TIMER_CMD_CLR	W SYNC	Command Register
0x2010	TIMER_STATUS_CLR	RH	Status Register
0x2014	TIMER_STATUS2_CLR	RH	Status Reg for Phase and Dither
0x2018	TIMER_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x201C	TIMER_IEN_CLR	RW	Interrupt Enable Register
0x2020	TIMER_TOP_CLR	RWH SYNC	Counter Top Value Register
0x2024	TIMER_TOPB_CLR	RW SYNC	Counter Top Value Buffer Register
0x2028	TIMER_CNT_CLR	RWH SYNC	Counter Value Register
0x202C	TIMER_LOCK_CLR	W	TIMER Configuration Lock Register
0x2030	TIMER_EN_CLR	RW ENABLE	Module En
0x2060	TIMER_CCx_CFG_CLR	RW CONFIG	CC Channel Configuration Register
0x2064	TIMER_CCx_CTRL_CLR	RW SYNC	CC Channel Control Register
0x2068	TIMER_CCx_PHASE_CLR	RWH SYNC	Channel Phase Register
0x206C	TIMER_CCx_PHASEB_CLR	RW SYNC	Channel Phase Buffer Register
0x2070	TIMER_CCx_OC_CLR	RWH SYNC	OC Channel Value Register

Offset	Name	Type	Description
0x2074	TIMER_CCx_OCB_CLR	RW SYNC	OC Channel Value Buffer Register
0x2078	TIMER_CCx_DITHER_CLR	RWH SYNC	Dither Value Register
0x207C	TIMER_CCx_DITHERB_CLR	RW SYNC	Dither Value Buffer Register
0x2084	TIMER_CCx_ICF_CLR	RH(r)	IC Channel Value Register
0x2088	TIMER_CCx_ICOF_CLR	RH SYNC	IC Channel Value Overflow Register
0x21B0	TIMER_DTCFG_CLR	RW CONFIG	DTI Configuration Register
0x21B4	TIMER_DTTIMECFG_CLR	RW CONFIG	DTI Time Configuration Register
0x21B8	TIMER_DTFCFG_CLR	RW CONFIG	DTI Fault Configuration Register
0x21BC	TIMER_DTCTRL_CLR	RW SYNC	DTI Control Register
0x21C0	TIMER_DTOGEN_CLR	RW SYNC	DTI Output Generation Enable Register
0x21C4	TIMER_DTFAULT_CLR	RH	DTI Fault Register
0x21C8	TIMER_DTFAULTC_CLR	W SYNC	DTI Fault Clear Register
0x21CC	TIMER_DTLOCK_CLR	W	DTI Configuration Lock Register
0x3000	TIMER_IPVERSION_TGL	R	IP Version ID
0x3004	TIMER_CFG_TGL	RW CONFIG	Configuration Register
0x3008	TIMER_CTRL_TGL	RW SYNC	Control Register
0x300C	TIMER_CMD_TGL	W SYNC	Command Register
0x3010	TIMER_STATUS_TGL	RH	Status Register
0x3014	TIMER_STATUS2_TGL	RH	Status Reg for Phase and Dither
0x3018	TIMER_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x301C	TIMER_IEN_TGL	RW	Interrupt Enable Register
0x3020	TIMER_TOP_TGL	RWH SYNC	Counter Top Value Register
0x3024	TIMER_TOPB_TGL	RW SYNC	Counter Top Value Buffer Register
0x3028	TIMER_CNT_TGL	RWH SYNC	Counter Value Register
0x302C	TIMER_LOCK_TGL	W	TIMER Configuration Lock Register
0x3030	TIMER_EN_TGL	RW ENABLE	Module En
0x3060	TIMER_CCx_CFG_TGL	RW CONFIG	CC Channel Configuration Register
0x3064	TIMER_CCx_CTRL_TGL	RW SYNC	CC Channel Control Register
0x3068	TIMER_CCx_PHASE_TGL	RWH SYNC	Channel Phase Register
0x306C	TIMER_CCx_PHASEB_TGL	RW SYNC	Channel Phase Buffer Register
0x3070	TIMER_CCx_OC_TGL	RWH SYNC	OC Channel Value Register
0x3074	TIMER_CCx_OCB_TGL	RW SYNC	OC Channel Value Buffer Register
0x3078	TIMER_CCx_DITHER_TGL	RWH SYNC	Dither Value Register
0x307C	TIMER_CCx_DITHERB_TGL	RW SYNC	Dither Value Buffer Register
0x3084	TIMER_CCx_ICF_TGL	RH(r)	IC Channel Value Register
0x3088	TIMER_CCx_ICOF_TGL	RH SYNC	IC Channel Value Overflow Register
0x31B0	TIMER_DTCFG_TGL	RW CONFIG	DTI Configuration Register
0x31B4	TIMER_DTTIMECFG_TGL	RW CONFIG	DTI Time Configuration Register
0x31B8	TIMER_DTFCFG_TGL	RW CONFIG	DTI Fault Configuration Register
0x31BC	TIMER_DTCTRL_TGL	RW SYNC	DTI Control Register
0x31C0	TIMER_DTOGEN_TGL	RW SYNC	DTI Output Generation Enable Register

Offset	Name	Type	Description
0x31C4	TIMER_DTFAULT_TGL	RH	DTI Fault Register
0x31C8	TIMER_DTFAULTC_TGL	W SYNC	DTI Fault Clear Register
0x31CC	TIMER_DTLOCK_TGL	W	DTI Configuration Lock Register

20.5 TIMER Register Description

20.5.1 TIMER_IPVERSION - IP Version ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x2																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x2	R	**IP Version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

20.5.2 TIMER_CFG - Configuration Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0										0x0	0x0			0x0	0x0	0x0	0x0	0x0		0x0	0x0	0x0	0x0	0x0		0x0	
Access					RW										RW	RW			RW	RW	RW	RW	RW		RW	RW	RW	RW	RW		RW	
Name					PRESC										RSSCOIST	ATI			UPDATEMODE	RETIMESEL	DISSYNCOUT	RETIMEEN	CLKSEL		DMACLRACT	DEBUGRUN	QDM	OSMEN	SYNC		MODE	

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:18	PRESC	0x0	RW	**Prescaler Setting**

These bits select the prescaling factor for the counter clock. The selected timer clock will be divided by PRESC+1 before clocking the counter. The following modes are provided for easier software porting from Series 0 or Series 1 devices. However, the prescaler is not limited to these options.

Value	Mode	Description
0	DIV1	No prescaling
1	DIV2	Prescale by 2
3	DIV4	Prescale by 4
7	DIV8	Prescale by 8
15	DIV16	Prescale by 16
31	DIV32	Prescale by 32
63	DIV64	Prescale by 64
127	DIV128	Prescale by 128
255	DIV256	Prescale by 256
511	DIV512	Prescale by 512
1023	DIV1024	Prescale by 1024

Bit	Name	Reset	Access	Description
17	RSSCOIST	0x0	RW	**Reload-Start Sets COIST**

When enabled, compare output is set to COIST value on a Reload-Start event.

Bit	Name	Reset	Access	Description
16	ATI	0x0	RW	**Always Track Inputs**

Enabling ATI makes CCPOL always track the polarity of the inputs.

Bit	Name	Reset	Access	Description
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	UPDATEMODE	0x0	RW	**Buffered output update mode**

Enable UPDATEMODE to control the writes from buffered (OCB, PHASEB, DITHERB) to direct (OC, DITHER, PHASE) registers.

Value	Mode	Description
0	ALL	Direct registers are updated from buffer on every applicable update event.
1	CMD	Direct registers are updated from buffer on the first applicable update event after CMD.UPDATECMD is set.

Bit	Name	Reset	Access	Description
12	RETIMESEL	0x0	RW	**PWM output retime select**
	When RETIMEEN is set, the PWM output stage will be re-timed to synchronize edges with radio clocks and reduce RF interference. This will introduce up to 1 cycle of clock jitter between PWM outputs.			
11	DISSYNCOUT	0x0	RW	**Disable Timer Start/Stop/Reload output**
	When this bit is set, the Timer does not start/stop/reload other timers with SYNC bit set.			

Value	Mode	Description
0	EN	Timer can start/stop/reload other timers with SYNC bit set
1	DIS	Timer cannot start/stop/reload other timers with SYNC bit set

Bit	Name	Reset	Access	Description
10	RETIMEEN	0x0	RW	**PWM output retimed enable**
	Enable retiming of PWM output.			

Value	Mode	Description
0	DISABLE	PWM outputs are not re-timed.
1	ENABLE	PWM outputs are re-timed.

Bit	Name	Reset	Access	Description
9:8	CLKSEL	0x0	RW	**Clock Source Select**
	These bits select the clock source for the timer.			

Value	Mode	Description
0	PRESCEM01GRPACLK	Prescaled EM01GRPACLK
1	CC1	Compare/Capture Channel 1 Input
2	TIMEROUF	Timer is clocked by underflow(down-count) or overflow(up-count) in the lower numbered neighbor Timer

Bit	Name	Reset	Access	Description
7	DMACLRACT	0x0	RW	**DMA Request Clear on Active**
	When this bit is set, the DMA requests are cleared when the corresponding DMA channel is active. This enables the timer DMA requests to be cleared without accessing the timer.			
6	DEBUGRUN	0x0	RW	**Debug Mode Run Enable**
	Set this bit to enable timer to run in debug mode.			

Value	Mode	Description
0	HALT	Timer is halted in debug mode
1	RUN	Timer is running in debug mode

Bit	Name	Reset	Access	Description
5	QDM	0x0	RW	**Quadrature Decoder Mode Selection**
	This bit sets the mode for the quadrature decoder.			

Value	Mode	Description
0	X2	X2 mode selected
1	X4	X4 mode selected

Bit	Name	Reset	Access	Description
4	OSMEN	0x0	RW	**One-shot Mode Enable**
	Enable/disable one shot mode.			
3	SYNC	0x0	RW	**Timer Start/Stop/Reload Synchronization**
	When this bit is set, the Timer is started/stopped/reloaded by start/stop/reload commands in the other timers.			

Value	Mode	Description

Bit	Name	Reset	Access	Description
	0	DISABLE		Timer operation is unaffected by other timers.
	1	ENABLE		Timer may be started, stopped and re-loaded from other timer instances.
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	MODE	0x0	RW	**Timer Mode**

These bits set the counting mode for the Timer. Note, when Quadrature Decoder Mode is selected (MODE = 'b11), the CLKSEL is don't care. The Timer is clocked by the Decoder Mode clock output.

Value	Mode	Description
0	UP	Up-count mode
1	DOWN	Down-count mode
2	UPDOWN	Up/down-count mode
3	QDEC	Quadrature decoder mode

20.5.3 TIMER_CTRL - Control Register

Offset	Bit Position																															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0		0x0	
Access																												RW	RW		RW	
Name																												X2CNT	FALLA		RISEA	

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	X2CNT	0x0	RW	**2x Count Mode**
	Enable 2x count mode			
3:2	FALLA	0x0	RW	**Timer Falling Input Edge Action**
	These bits select the action taken in the counter when a falling edge occurs on the input.			

Value	Mode	Description
0	NONE	No action
1	START	Start counter without reload
2	STOP	Stop counter without reload
3	RELOADSTART	Reload and start counter

1:0	RISEA	0x0	RW	**Timer Rising Input Edge Action**
	These bits select the action taken in the counter when a rising edge occurs on the input.			

Value	Mode	Description
0	NONE	No action
1	START	Start counter without reload
2	STOP	Stop counter without reload
3	RELOADSTART	Reload and start counter

20.5.4 TIMER_CMD - Command Register

Offset	Bit Position																														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														W(nB)	W(nB)	W(nB)
Name																														UPDATECMD	STOP	START

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	UPDATECMD	0x0	W(nB)	**Buffered output update**
	This bit is set to trigger update of buffered output registers when CFG.UPDATEMODE is set to CMD			
1	STOP	0x0	W(nB)	**Stop Timer**
	This bit is set to stop timer using software			
0	START	0x0	W(nB)	**Start Timer**
	This bit is set to start timer using software			

20.5.5 TIMER_STATUS - Status Register

Offset	Bit Position																														
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x0	0x0	0x0						0x0	0x0	0x0						0x0	0x0	0x0		0x0	0x0	0x0		0x0	0x0	0x0
Access						R	R	R						R	R	R						R	R	R		R	R	R		R	R	R
Name						CCPOL2	CCPOL1	CCPOL0						ICFEMPTY2	ICFEMPTY1	ICFEMPTY0						OCBV2	OCBV1	OCBV0		SYNCBUSY	DTILOCKSTATUS	TIMERLOCKSTATUS		TOPBV	DIR	RUNNING

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26	CCPOL2	0x0	R	**Compare/Capture Polarity**

In Input Capture mode, this bit indicates the polarity of the edge that triggered capture in TIMERn_CCx_CCV. In Compare/PWM mode, this bit indicates the polarity of the selected input to CC channel x. These bits are cleared when CCMODE is written to 0b00 (Off).

Value	Mode	Description
0	LOWRISE	CCx polarity low level/rising edge
1	HIGHFALL	CCx polarity high level/falling edge

25	CCPOL1	0x0	R	**Compare/Capture Polarity**

In Input Capture mode, this bit indicates the polarity of the edge that triggered capture in TIMERn_CCx_CCV. In Compare/PWM mode, this bit indicates the polarity of the selected input to CC channel x. These bits are cleared when CCMODE is written to 0b00 (Off).

Value	Mode	Description
0	LOWRISE	CCx polarity low level/rising edge
1	HIGHFALL	CCx polarity high level/falling edge

24	CCPOL0	0x0	R	**Compare/Capture Polarity**

In Input Capture mode, this bit indicates the polarity of the edge that triggered capture in TIMERn_CCx_CCV. In Compare/PWM mode, this bit indicates the polarity of the selected input to CC channel x. These bits are cleared when CCMODE is written to 0b00 (Off).

Value	Mode	Description
0	LOWRISE	CCx polarity low level/rising edge
1	HIGHFALL	CCx polarity high level/falling edge

23:19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18	ICFEMPTY2	0x0	R	**Input capture fifo empty**

Set when input capture FIFO is empty

17	ICFEMPTY1	0x0	R	**Input capture fifo empty**

Set when input capture FIFO is empty

16	ICFEMPTY0	0x0	R	**Input capture fifo empty**

Bit	Name	Reset	Access	Description
	Set when input capture FIFO is empty			
15:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	OCBV2	0x0	R	**Output Compare Buffer Valid**
	This field indicates that the TIMERn_CCx_OCB register contain data which has not been written to TIMERn_CCx_OC. These bits are only used in OUTPUTCOMPARE or PWM mode and are cleared when CCMODE is written to 0b00 (Off).			
9	OCBV1	0x0	R	**Output Compare Buffer Valid**
	This field indicates that the TIMERn_CCx_OCB register contain data which has not been written to TIMERn_CCx_OC. These bits are only used in OUTPUTCOMPARE or PWM mode and are cleared when CCMODE is written to 0b00 (Off).			
8	OCBV0	0x0	R	**Output Compare Buffer Valid**
	This field indicates that the TIMERn_CCx_OCB register contain data which has not been written to TIMERn_CCx_OC. These bits are only used in OUTPUTCOMPARE or PWM mode and are cleared when CCMODE is written to 0b00 (Off).			
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	SYNCBUSY	0x0	R	**Sync Busy**
	Indicates synchronization ongoing			
5	DTILOCKSTATUS	0x0	R	**DTI lock status**
	Indicates current status of DTI lock			

Value	Mode	Description
0	UNLOCKED	DTI registers are unlocked
1	LOCKED	DTI registers are locked

4	TIMERLOCKSTATUS	0x0	R	**Timer lock status**
	Indicates current status of Timer lock			

Value	Mode	Description
0	UNLOCKED	TIMER registers are unlocked
1	LOCKED	TIMER registers are locked

3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	TOPBV	0x0	R	**TOP Buffer Valid**
	This indicates that TIMERn_TOPB contains valid data that has not been written to TIMERn_TOP. This bit is also cleared when TIMERn_TOP is written.			
1	DIR	0x0	R	**Direction**
	Indicates count direction.			

Value	Mode	Description
0	UP	Counting up
1	DOWN	Counting down

0	RUNNING	0x0	R	**Running**
	Indicates if timer is running or not.			

20.5.6 TIMER_STATUS2 - Status Reg for Phase and Dither

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																0x0						0x0	0x0	0x0						0x0	0x0	0x0
Access																R						R	R	R						R	R	R
Name																UPDATEBUSY						DITHERBV2	DITHERBV1	DITHERBV0						PHASEBV2	PHASEBV1	PHASEBV0

Bit	Name	Reset	Access	Description
31:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	UPDATEBUSY	0x0	R	**Buffer output update busy**
	This bit is set when UPDATECMD is issued, and cleared after an update event when the buffered values have been updated in the direct registers.			
15:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	DITHERBV2	0x0	R	**Dither valid buffer**
	This field indicates that the TIMERn_CCx_DITHERB register contains valid data (data which has not been written to TIMERn_CCx_DITHER yet). This bit only used in PWM mode and is cleared once the DITHERB buffer value is written to DITHER reg or when CCMODE is written to 0b00 (Off).			
9	DITHERBV1	0x0	R	**Dither valid buffer**
	This field indicates that the TIMERn_CCx_DITHERB register contains valid data (data which has not been written to TIMERn_CCx_DITHER yet). This bit only used in PWM mode and is cleared once the DITHERB buffer value is written to DITHER reg or when CCMODE is written to 0b00 (Off).			
8	DITHERBV0	0x0	R	**Dither valid buffer**
	This field indicates that the TIMERn_CCx_DITHERB register contains valid data (data which has not been written to TIMERn_CCx_DITHER yet). This bit only used in PWM mode and is cleared once the DITHERB buffer value is written to DITHER reg or when CCMODE is written to 0b00 (Off).			
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	PHASEBV2	0x0	R	**Phase valid buffer**
	This field indicates that the TIMERn_CCx_PHASEB register contain valid data (data which has not been written to TIMERn_CCx_PHASE reg yet). This bit is only used in PWM mode and is cleared once the PHASEB buffer value is written to PHASE reg or when CCMODE is written to 0b00 (Off).			
1	PHASEBV1	0x0	R	**Phase valid buffer**
	This field indicates that the TIMERn_CCx_PHASEB register contain valid data (data which has not been written to TIMERn_CCx_PHASE reg yet). This bit is only used in PWM mode and is cleared once the PHASEB buffer value is written to PHASE reg or when CCMODE is written to 0b00 (Off).			
0	PHASEBV0	0x0	R	**Phase valid buffer**
	This field indicates that the TIMERn_CCx_PHASEB register contain valid data (data which has not been written to TIMERn_CCx_PHASE reg yet). This bit is only used in PWM mode and is cleared once the PHASEB buffer value is written to PHASE reg or when CCMODE is written to 0b00 (Off).			

20.5.7 TIMER_IF - Interrupt Flag Register

Offset	Bit Position																														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0	0x0	0x0					0x0	0x0	0x0					0x0	0x0	0x0					0x0	0x0	0x0		0x0	0x0	0x0
Access					RW	RW	RW					RW	RW	RW					RW	RW	RW					RW	RW	RW		RW	RW	RW
Name					ICFUF2	ICFUF1	ICFUF0					ICFOF2	ICFOF1	ICFOF0					ICFWLFULL2	ICFWLFULL1	ICFWLFULL0					CC2	CC1	CC0		DIRCHG	UF	OF

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	ICFUF2	0x0	RW	**Input capture FIFO underflow**
	Indicates that software tried to read an empty FIFO on channel 2.			
26	ICFUF1	0x0	RW	**Input capture FIFO underflow**
	Indicates that software tried to read an empty FIFO on channel 1.			
25	ICFUF0	0x0	RW	**Input capture FIFO underflow**
	Indicates that software tried to read an empty FIFO on channel 0.			
24:21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	ICFOF2	0x0	RW	**Input Capture FIFO overflow**
	Indicates that input capture FIFO for channel 2 has overflown, and a prior captured value was lost. The latest captured value can be read from the ICOF register.			
19	ICFOF1	0x0	RW	**Input Capture FIFO overflow**
	Indicates that input capture FIFO for channel 1 has overflown, and a prior captured value was lost. The latest captured value can be read from the ICOF register.			
18	ICFOF0	0x0	RW	**Input Capture FIFO overflow**
	Indicates that input capture FIFO for channel 0 has overflown, and a prior captured value was lost. The latest captured value can be read from the ICOF register.			
17:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	ICFWLFULL2	0x0	RW	**Input Capture Watermark Level Full**
	This bit indicates that the Input capture FIFO watermark for channel 2 has been exceeded.			
12	ICFWLFULL1	0x0	RW	**Input Capture Watermark Level Full**
	This bit indicates that the Input capture FIFO watermark for channel 1 has been exceeded.			
11	ICFWLFULL0	0x0	RW	**Input Capture Watermark Level Full**
	This bit indicates that the Input capture FIFO watermark for channel 0 has been exceeded.			
10:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	CC2	0x0	RW	**Capture Compare Channel 2 Interrupt Flag**
	In INPUT CAPTURE mode this bit indicates that a new Capture event has taken place. In OUTPUTCOMPARE or PWM mode this bit indicates that a match event has taken place			
5	CC1	0x0	RW	**Capture Compare Channel 1 Interrupt Flag**

Bit	Name	Reset	Access	Description
	In INPUT CAPTURE mode this bit indicates that a new Capture event has taken place. In OUTPUTCOMPARE or PWM mode this bit indicates that a match event has taken place			
4	CC0	0x0	RW	**Capture Compare Channel 0 Interrupt Flag**
	In INPUT CAPTURE mode this bit indicates that a new Capture event has taken place. In OUTPUTCOMPARE or PWM mode this bit indicates that a match event has taken place			
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	DIRCHG	0x0	RW	**Direction Change Detect Interrupt Flag**
	This bit is set when count direction changes. Set only in Quadrature Decoder mode			
1	UF	0x0	RW	**Underflow Interrupt Flag**
	This bit indicates that there has been an underflow.(counter value reaches 0 in downcounting mode)			
0	OF	0x0	RW	**Overflow Interrupt Flag**
	This bit indicates that there has been an overflow.(counter value reaches the max value(TOP) in up count mode)			

20.5.8 TIMER_IEN - Interrupt Enable Register

Offset	Bit Position																														
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0	0x0	0x0					0x0	0x0	0x0					0x0	0x0	0x0					0x0	0x0	0x0		0x0	0x0	0x0
Access					RW	RW	RW					RW	RW	RW					RW	RW	RW					RW	RW	RW		RW	RW	RW
Name					ICFUF2	ICFUF1	ICFUF0					ICFOF2	ICFOF1	ICFOF0					ICFWLFULL2	ICFWLFULL1	ICFWLFULL0					CC2	CC1	CC0		DIRCHG	UF	OF

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	ICFUF2	0x0	RW	**ICFUF2 Interrupt Enable**
	Enable/Disable the ICFUF2 interrupt			
26	ICFUF1	0x0	RW	**ICFUF1 Interrupt Enable**
	Enable/Disable the ICFUF1 interrupt			
25	ICFUF0	0x0	RW	**ICFUF0 Interrupt Enable**
	Enable/Disable the ICFUF0 interrupt			
24:21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	ICFOF2	0x0	RW	**ICFOF2 Interrupt Enable**
	Enable/Disable the ICFOF2 interrupt			
19	ICFOF1	0x0	RW	**ICFOF1 Interrupt Enable**
	Enable/Disable the ICFOF1 interrupt			
18	ICFOF0	0x0	RW	**ICFOF0 Interrupt Enable**
	Enable/Disable the ICFOF0 interrupt			
17:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	ICFWLFULL2	0x0	RW	**ICFWLFULL2 Interrupt Enable**
	Enable/Disable the ICFWLFULL2 interrupt			
12	ICFWLFULL1	0x0	RW	**ICFWLFULL1 Interrupt Enable**
	Enable/Disable the ICFWLFULL1 interrupt			
11	ICFWLFULL0	0x0	RW	**ICFWLFULL0 Interrupt Enable**
	Enable/Disable the ICFWLFULL0 interrupt			
10:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	CC2	0x0	RW	**CC2 Interrupt Enable**
	Enable/Disable the CC2 interrupt			
5	CC1	0x0	RW	**CC1 Interrupt Enable**
	Enable/Disable the CC1 interrupt			
4	CC0	0x0	RW	**CC0 Interrupt Enable**
	Enable/Disable the CC0 interrupt			

Bit	Name	Reset	Access	Description
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	DIRCHG	0x0	RW	**Direction Change Detect Interrupt Enable**
	Disable the DIRCHG interrupt			
1	UF	0x0	RW	**Underflow Interrupt Enable**
	Enable/Disable the UF interrupt			
0	OF	0x0	RW	**Overflow Interrupt Enable**
	Enable/Disable the OF interrupt			

20.5.9 TIMER_TOP - Counter Top Value Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0xFFFF																															
Access	RW																															
Name	TOP																															

Bit	Name	Reset	Access	Description
31:0	TOP	0xFFFF	RW	**Counter Top Value**
	this reg holds the top value- the maximum value the counter should count to, in the timer			

20.5.10 TIMER_TOPB - Counter Top Value Buffer Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	TOPB																															

Bit	Name	Reset	Access	Description
31:0	TOPB	0x0	RW	**Counter Top Buffer Register**
	This is the TOP buffer register which is used to write to the TOP register for on the fly glitch-less update to TOP reg. it is written to TOP only on an update event if TOPB contains valid data			

20.5.11 TIMER_CNT - Counter Value Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	CNT																															

Bit	Name	Reset	Access	Description
31:0	CNT	0x0	RW	**Counter Value**
	These bits hold the counter value.			

20.5.12 TIMER_LOCK - TIMER Configuration Lock Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0x0	W	**Timer Lock Key**
	Write any other value than the unlock code to lock TIMERn_CTRL, TIMERn_CFG, TIMERn_CMD, TIMERn_TOP, TIMERn_CNT, TIMERn_CCx_CTRL, TIMERn_CCx_CFG, and TIMERn_CCx_OC from editing. Write the unlock code to unlock these registers.			

Value	Mode	Description
52864	UNLOCK	Write to unlock TIMER registers

20.5.13 TIMER_EN - Module En

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status** When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFO
0	EN	0x0	RW	**Timer Module Enable** The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.

20.5.14 TIMER_CCx_CFG - CC Channel Configuration Register

Offset	Bit Position																															
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0	0x0	0x0	0x0														0x0			0x0	
Access											RW	RW	RW	RW														RW			RW	
Name											ICFWL	FILT	PRSCONF	INSEL														COIST			MODE	

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	ICFWL	0x0	RW	**Input Capture FIFO watermark level**
	Sets the watermark level for generation of the ICFWLFULL interrupt and DMA requests. ICFWLFULL will be set and DMA requests may be generated if the number of free FIFO entries is less than or equal to ICFWL.			
20	FILT	0x0	RW	**Digital Filter**
	Enable digital filter.			

Value	Mode	Description
0	DISABLE	Digital Filter Disabled
1	ENABLE	Digital Filter Enabled

Bit	Name	Reset	Access	Description
19	PRSCONF	0x0	RW	**PRS Configuration**
	Select PRS pulse or level for PRS output.			

Value	Mode	Description
0	PULSE	Each CC event will generate a one EM01GRPACLK cycle high pulse
1	LEVEL	The PRS channel will follow CC out

Bit	Name	Reset	Access	Description
18:17	INSEL	0x0	RW	**Input Selection**
	Select Compare/Capture channel input.			

Value	Mode	Description
0	PIN	TIMERnCCx pin is selected
1	PRSSYNC	Synchornous PRS selected
2	PRSASYNCLEVEL	Asynchronous Level PRS selected
3	PRSASYNCPULSE	Asynchronous Pulse PRS selected

Bit	Name	Reset	Access	Description
16:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	COIST	0x0	RW	**Compare Output Initial State**
	This bit is only used in Output Compare and PWM mode. When this bit is set in Compare or PWM mode,the output is set high when the counter is disabled. When counting resumes, this value will represent the initial value for the output. If the bit is cleared, the output will be cleared when the counter is disabled.			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	MODE	0x0	RW	**CC Channel Mode**

Bit	Name	Reset	Access	Description

These bits select the mode for Compare/Capture channel.

Value	Mode	Description
0	OFF	Compare/Capture channel turned off
1	INPUTCAPTURE	Input Capture
2	OUTPUTCOMPARE	Output Compare
3	PWM	Pulse-Width Modulation

20.5.15 TIMER_CCx_CTRL - CC Channel Control Register

Offset	Bit Position																													
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0		0x0												0x0		0x0		0x0							0x0		
Access					RW		RW												RW		RW		RW							RW		
Name					ICEVCTRL		ICEDGE												CUFOA		COFOA		CMOA							OUTINV		

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:26	ICEVCTRL	0x0	RW	**Input Capture Event Control**

These bits control when a Compare/Capture PRS output pulse and interrupt flag is set. DMA request however is set on every capture.

Value	Mode	Description
0	EVERYEDGE	PRS output pulse and interrupt flag set on every capture
1	EVERYSECONDEDGE	PRS output pulse and interrupt flag set on every second capture
2	RISING	PRS output pulse and interrupt flag set on rising edge only (if ICEDGE = BOTH)
3	FALLING	PRS output pulse and interrupt flag set on falling edge only (if ICEDGE = BOTH)

25:24	ICEDGE	0x0	RW	**Input Capture Edge Select**

These bits control which edges the edge detector triggers on. The output is used for input capture and external clock input.

Value	Mode	Description
0	RISING	Rising edges detected
1	FALLING	Falling edges detected
2	BOTH	Both edges detected
3	NONE	No edge detection, signal is left as it is

23:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	CUFOA	0x0	RW	**Counter Underflow Output Action**

Select output action on counter underflow.

Value	Mode	Description
0	NONE	No action on counter underflow
1	TOGGLE	Toggle output on counter underflow
2	CLEAR	Clear output on counter underflow
3	SET	Set output on counter underflow

11:10	COFOA	0x0	RW	**Counter Overflow Output Action**

Select output action on counter overflow.

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	NONE		No action on counter overflow
	1	TOGGLE		Toggle output on counter overflow
	2	CLEAR		Clear output on counter overflow
	3	SET		Set output on counter overflow
9:8	CMOA	0x0	RW	**Compare Match Output Action**
	Select output action on compare match.			
	Value	Mode		Description
	0	NONE		No action on compare match
	1	TOGGLE		Toggle output on compare match
	2	CLEAR		Clear output on compare match
	3	SET		Set output on compare match
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	OUTINV	0x0	RW	**Output Invert**
	Setting this bit inverts the output from the CC channel (Output compare or PWM mode).			
1:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

20.5.16 TIMER_CCx_PHASE - Channel Phase Register

Offset	Bit Position																													
0x068	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	PHASE																															

Bit	Name	Reset	Access	Description
31:0	PHASE	0x0	RW	**Phase Value of CC**
	The PHASE reg value determines the value at which cc output is set to 1 in PWM upcount mode. It can be used to delay the start of cc_out(cc_out is set to 1 at cnt=1 in a legacy timer). Set to 0 to function like a legacy timer			

20.5.17 TIMER_CCx_PHASEB - Channel Phase Buffer Register

Offset	Bit Position																													
0x06C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	PHASEB																															

Bit	Name	Reset	Access	Description
31:0	PHASEB	0x0	RW	**Phase Value Buffer**
	Dither buffer register is used for on the fly glitchless update of PHASE register.The PHASEB value will be written to TIMERn_CCx_PHASE on an update event if TIMERn_CCx_PHASEB contains valid data.			

20.5.18 TIMER_CCx_OC - OC Channel Value Register

Offset	Bit Position																													
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	OC																															

Bit	Name	Reset	Access	Description
31:0	OC	0x0	RW	**Output Compare Value**
	This fields holds the output compare value			

20.5.19 TIMER_CCx_OCB - OC Channel Value Buffer Register

Offset	Bit Position																													
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	OCB																															

Bit	Name	Reset	Access	Description
31:0	OCB	0x0	RW	**Output Compare Value Buffer**
	This field holds the Output Compare buffer value which will be written to TIMERn_CCx_OC on an update event if TIMERn_CCx_OCB contains valid data			

20.5.20 TIMER_CCx_DITHER - Dither Value Register

Offset	Bit Position																															
0x078	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													DITHER			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	DITHER	0x0	RW	**Dither Value** The dither reg value determines the number of pwm pulses to add the dithering to, out of 2^DITHERWIDTH pulses.(Dithering makes a PWM pulse output one clock cycle wider)

20.5.21 TIMER_CCx_DITHERB - Dither Value Buffer Register

Offset	Bit Position																															
0x07C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													DITHERB			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	DITHERB	0x0	RW	**Dither Value Buffer** Dither buffer register is used for on the fly glitchless update of DITHER register.The DITHERB value will be written to TIMERn_CCx_DITHER on an update event if TIMERn_CCx_DITHERB contains valid data

20.5.22 TIMER_CCx_ICF - IC Channel Value Register

Offset	Bit Position																															
0x084	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R(r)																															
Name	ICF																															

Bit	Name	Reset	Access	Description
31:0	ICF	0x0	R(r)	**Input Capture FIFO** This FIFO holds captured values in input capture mode. Reading this register will pop the oldest unread value from the FIFO.

20.5.23 TIMER_CCx_ICOF - IC Channel Value Overflow Register

Offset	Bit Position																													
0x088	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	ICOF																															

Bit	Name	Reset	Access	Description
31:0	ICOF	0x0	R	**Input Capture FIFO Overflow**
	This register always contains the most recent input capture value. If the input capture FIFO is full and a new capture occurs, this register will be updated and the previous caputre value is over-written.			

20.5.24 TIMER_DTCFG - DTI Configuration Register

Offset	Bit Position																													
0x1B0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0								0x0	0x0
Access																					RW	RW	RW								RW	RW
Name																					DTPRSEN	DTFATS	DTAR								DTDAS	DTEN

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	DTPRSEN	0x0	RW	**DTI PRS Source Enable**
	Enable/disable PRS as DTI input.			
10	DTFATS	0x0	RW	**DTI Fault Action on Timer Stop**
	When Timer stops, DTI block outputs go to safe state as programmed in DTFA field of TIMERn_DTFC register. However, when DTAR is also set,DTAR having higher priority allows channel0 to output the incoming PRS input while the other channels go to safe state			
9	DTAR	0x0	RW	**DTI Always Run**
	This is used only for DTI channel 0. It Allows DTI channel 0 to keep running even when the timer is stopped. This is useful when its input source is PRS. However, here the undivided peripheral clock is always used regardless of the programmed value in DTPRESC.			
8:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DTDAS	0x0	RW	**DTI Automatic Start-up Functionality**
	Configure DTI restart on debugger exit.			

Value	Mode	Description
0	NORESTART	No DTI restart on debugger exit
1	RESTART	DTI restart on debugger exit

0	DTEN	0x0	RW	**DTI Enable**
	Enable/disable Dead time insertion			

20.5.25 TIMER_DTTIMECFG - DTI Time Configuration Register

Offset	Bit Position																													
0x1B4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0						0x0						0x0									
Access											RW						RW						RW									
Name											DTFALLT						DTRISET						DTPRESC									

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:16	DTFALLT	0x0	RW	**DTI Fall-time**
	Set time span for the falling edge. The fall time is DTFALLT+1 prescaled peripheral clock cycles			
15:10	DTRISET	0x0	RW	**DTI Rise-time**
	Set time span for the rising edge. The rise time is DTRISET+1 prescaled peripheral clock cycles			
9:0	DTPRESC	0x0	RW	**DTI Prescaler Setting**
	These bits select the prescaling factor for DTI. The selected timer clock will be divided by DTPRESC+1 before clocking the DTI logic.			

20.5.26 TIMER_DTFCFG - DTI Fault Configuration Register

Offset	Bit Position																													
0x1B8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0	0x0	0x0	0x0							0x0																	
Access					RW	RW	RW	RW							RW																	
Name					DTLOCKUPFEN	DTDBGFEN	DTPRS1FEN	DTPRS0FEN							DTFA																	

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	DTLOCKUPFEN	0x0	RW	**DTI Lockup Fault Enable**
	Set this bit to 1 to enable core lockup as a fault source			
26	DTDBGFEN	0x0	RW	**DTI Debugger Fault Enable**
	Set this bit to 1 to enable debugger as a fault source			
25	DTPRS1FEN	0x0	RW	**DTI PRS 1 Fault Enable**
	Set this bit to 1 to enable PRS source 1 as a fault source			
24	DTPRS0FEN	0x0	RW	**DTI PRS 0 Fault Enable**
	Set this bit to 1 to enable PRS source 0 as a fault source			
23:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	DTFA	0x0	RW	**DTI Fault Action**
	Select fault action.			

Value	Mode	Description
0	NONE	No action on fault
1	INACTIVE	Set outputs inactive
2	CLEAR	Clear outputs
3	TRISTATE	Tristate outputs

15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

20.5.27 TIMER_DTCTRL - DTI Control Register

Offset	Bit Position																													
0x1BC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															DTIPOL	DTCINV

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DTIPOL	0x0	RW	**DTI Inactive Polarity**
	Set inactive polarity of outputs			
0	DTCINV	0x0	RW	**DTI Complementary Output Invert.**
	DTI Complementary Output Invert.			

20.5.28 TIMER_DTOGEN - DTI Output Generation Enable Register

Offset	Bit Position																													
0x1C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											DTOGCDTI2EN	DTOGCDTI1EN	DTOGCDTI0EN	DTOGCC2EN	DTOGCC1EN	DTOGCC0EN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	DTOGCDTI2EN	0x0	RW	**DTI CDTIn Output Generation Enable**
	This bit enables/disables output generation for the CDTI output from the DTI.			
4	DTOGCDTI1EN	0x0	RW	**DTI CDTIn Output Generation Enable**
	This bit enables/disables output generation for the CDTI output from the DTI.			
3	DTOGCDTI0EN	0x0	RW	**DTI CDTIn Output Generation Enable**
	This bit enables/disables output generation for the CDTI output from the DTI.			
2	DTOGCC2EN	0x0	RW	**DTI CCn Output Generation Enable**
	This bit enables/disables output generation for the CC output from the DTI.			
1	DTOGCC1EN	0x0	RW	**DTI CCn Output Generation Enable**
	This bit enables/disables output generation for the CC output from the DTI.			
0	DTOGCC0EN	0x0	RW	**DTI CCn Output Generation Enable**
	This bit enables/disables output generation for the CC output from the DTI.			

20.5.29 TIMER_DTFAULT - DTI Fault Register

Offset	Bit Position																															
0x1C4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													R	R	R	R
Name																													DTLOCKUPF	DTDBGF	DTPRS1F	DTPRS0F

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	DTLOCKUPF	0x0	R	**DTI Lockup Fault** This bit is set to 1 if a core lockup fault has occurred and DTLOCKUPFEN is set to 1. The TIMER0_DTFAULTC register can be used to clear fault bits.
2	DTDBGF	0x0	R	**DTI Debugger Fault** This bit is set to 1 if a debugger fault has occurred and DTDBGFEN is set to 1. The TIMER0_DTFAULTC register can be used to clear fault bits.
1	DTPRS1F	0x0	R	**DTI PRS 1 Fault** This bit is set to 1 if a PRS 1 fault has occurred and DTPRS1FEN is set to 1. The TIMER0_DTFAULTC register can be used to clear fault bits.
0	DTPRS0F	0x0	R	**DTI PRS 0 Fault** This bit is set to 1 if a PRS 0 fault has occurred and DTPRS0FEN is set to 1. The TIMER0_DTFAULTC register can be used to clear fault bits.

20.5.30 TIMER_DTFAULTC - DTI Fault Clear Register

Offset	Bit Position																															
0x1C8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													W(nB)	W(nB)	W(nB)	W(nB)
Name																													DTLOCKUPFC	DTDBGFC	DTPRS1FC	DTPRS0FC

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	DTLOCKUPFC	0x0	W(nB)	**DTI Lockup Fault Clear**
	Write 1 to this bit to clear core lockup fault.			
2	DTDBGFC	0x0	W(nB)	**DTI Debugger Fault Clear**
	Write 1 to this bit to clear debugger fault.			
1	DTPRS1FC	0x0	W(nB)	**DTI PRS1 Fault Clear**
	Write 1 to this bit to clear PRS 1 fault.			
0	DTPRS0FC	0x0	W(nB)	**DTI PRS0 Fault Clear**
	Write 1 to this bit to clear PRS 0 fault.			

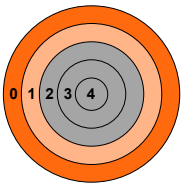
20.5.31 TIMER_DTLOCK - DTI Configuration Lock Register

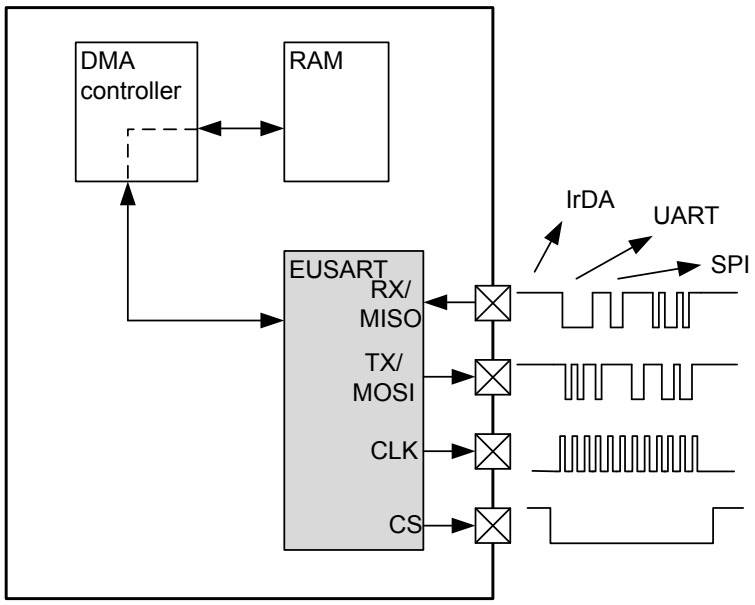
Offset	Bit Position																															
0x1CC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	W															
Name																	DTILOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	DTILOCKKEY	0x0	W	**DTI Lock Key**
	Write any other value than the unlock code to lock TIMER_ROUTE, TIMER_DTCTRL, TIMER_DTCFG, TIMER_DTTIMECFG and TIMER_DTFCFG from editing. Write the unlock code to unlock the DTI registers.			

Value	Mode	Description
52864	UNLOCK	Write to unlock TIMER DTI registers

21. EUSART - Universal Synchronous Asynchronous Receiver/Transmitter





Quick Facts

What?

The EUSART handles high-speed UART, SPI-bus, and IrDA communication.

Why?

Serial communication is frequently used in embedded systems and the EUSART allows efficient communication with a wide range of external devices.

How?

The EUSART has a wide selection of operating modes, frame formats, and baud rates. The multiprocessor mode allows the EUSART to remain idle when not addressed. 32-deep FIFOs and DMA support makes high data rates possible with minimal CPU intervention and it is possible to transmit and receive large frames while the MCU remains in EM1 Sleep.

21.1 Introduction

The Enhanced Universal Synchronous Asynchronous serial Receiver and Transmitter (EUSART) is a very flexible serial I/O module. It supports full duplex asynchronous UART communication as well as SPI, MicroWire. It can also interface with IrDA devices.

21.2 Features

- Asynchronous (UART) and synchronous (SPI) communication
- Full duplex and half duplex
- Separate TX/RX enable
- Separate receive / transmit 32 deep FIFOs, with additional separate shift registers
- Can operate using HF clock or LF clock (EUSART0)
- Programmable baud rate, generated as an fractional division from the peripheral clock
- Max bit-rate
  - Main SPI mode: 20 MHz
  - Secondary SPI mode: 10 MHz
  - UART mode, HF peripheral clock rate/16, 8, 6, or 4
  - UART mode, 32.768 kHz LF peripheral clock: 9600 (EUSART0, oversampling not supported with LF operation)
- Supports transmission and reception with
  - Full DMA support
  - UART mode, specified start-frame can start reception automatically
- UART mode
  - Either wake-up on any received byte or
  - Wake up only on specified start and signal frames
- SPI mode, capable of sleep-mode wake-up on received frame
  - Either wake-up on CS active or
  - Wake up on Receive FIFO level matching Watermark Level Setting
- Asynchronous mode supports
  - Configurable number of data bits 7-9 (plus the parity bit, if enabled)
  - HW parity bit generation and check
  - Majority vote baud-reception
  - False start-bit detection
  - Break generation/detection
  - Multi-processor mode
  - Configurable number of stop bits in asynchronous mode:
    - HF clock operation: 0.5, 1, 1.5, 2
    - LF clock operation: 1, 2 (EUSART0)
  - HW collision detection
  - IrDA support
    - HF clock operation: IrDA modulator
    - LF clock operation: Pulse extender, RX-only (EUSART0)
  - Hardware Flow Control
  - Automatic Baud Rate Detection
- Synchronous mode supports
  - Configurable number of data bits 8-16
  - All 4 SPI clock polarity/phase configurations
  - Main and Secondary interface modes
- Data can be transmitted LSB first or MSB first
- Separate interrupt vectors for receive and transmit interrupts
- Loopback mode
  - Half duplex communication
  - Communication debugging
- PRS RX input
- DMA Support
- Low energy operation with LF clock (EUSART0), with DMA interaction
- Async mode Automatic Baud Rate Detection operating with HF clock

21.3 Functional Description

An overview of the EUSART module is shown in Figure 21.1. EUSART Overview.

This section describes all posible EUSART features. Please refer to the Device Data Sheet to see what features a specific EUSART instance supports.

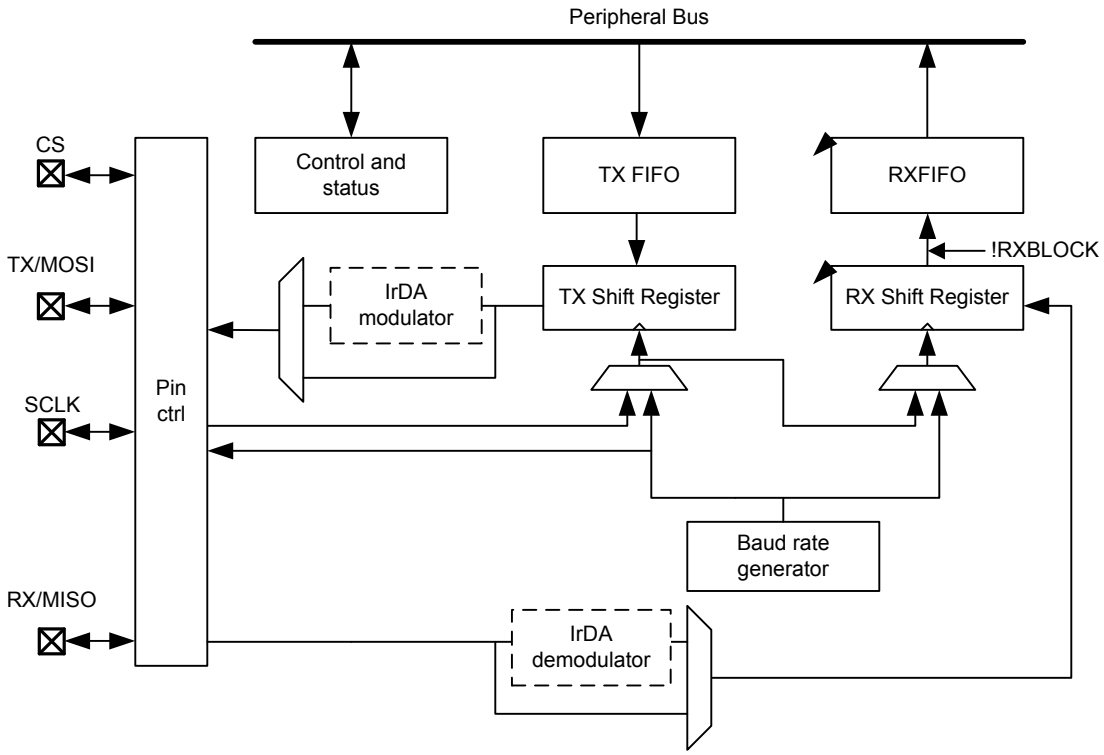


Figure 21.1. EUSART Overview

21.3.1 Power Domains and Energy Modes

Individual EUSART instances are powered from different power domains in the device, which may impact clocking options and availability in different energy modes. See 8. CMU - Clock Management Unit for more details on the clocking. The power domain and energy mode availability for each instance are summarized in the following table.

Table 21.1. EUSART Power Domains

Instance	Power Domain	Energy Modes
EUSART0	PD0D	EM0, EM1
EUSART1	PD1	EM0, EM1
EUSART2	PD1	EM0, EM1

21.3.2 Modes of Operation

The EUSART operates in either asynchronous or synchronous mode.

In synchronous mode, a separate clock signal is transmitted with the data. This clock signal is generated by the main interface on the bus, and both the main and secondary devices sample and transmit data according to this clock. Both main and secondary interface modes are supported by the EUSART. The synchronous communication mode is compatible with the Serial Peripheral Interface Bus (SPI) standard.

In asynchronous mode, no separate clock signal is transmitted with the data on the bus. The EUSART receiver thus has to determine where to sample the data on the bus from the actual data. To make this possible, additional synchronization bits are added to the data when operating in asynchronous mode, resulting in a slight overhead.

Asynchronous or synchronous mode can be selected by configuring SYNC in EUSARTn_CFG0. The options are listed with supported protocols in Table 21.2. EUSART Asynchronous vs. Synchronous Mode. Full duplex and half duplex communication is supported in both asynchronous and synchronous mode.

Table 21.2. EUSART Asynchronous vs. Synchronous Mode

SYNC	Communication Mode	Supported Protocols
0	Asynchronous	RS-232, IrDA
1	Synchronous	SPI, MicroWire

Table 21.3. EUSART Pin Usage explains the functionality of the different EUSART pins when the EUSART operates in different modes. Pin functionality enclosed in square brackets is optional, and depends on additional configuration parameters. LOOPBK and MASTER are discussed in 21.3.3.7 Local Loopback and 21.3.4.3 Main SPI Interface Mode (Clock Driver) respectively.

Table 21.3. EUSART Pin Usage

SYNC	LOOPBK	MASTER	Pin Functionality			
			TX (MOSI)	RX (MISO)	CLK	CS
0	0	x	Data out	Data in	-	-
0	1	x	Data out/in	-	-	-
1	0	0	Data in	Data out	Clock in	Secondary select
1	0	1	Data out	Data in	Clock out	[Auto secondary select]
1	1	0	Data out/in	-	Clock in	Secondary select
1	1	1	Data out/in	-	Clock out	[Auto secondary select]

21.3.3 Asynchronous Operation

EUSART can operate in asynchronous mode when EUSARTn_CFG0.SYNC is set to 0.

EUSART0 may operate as either a high-speed peripheral running from a high-frequency clock source (HF mode), or as a low-energy peripheral operating from a low-frequency clock source (LF mode). EUSART0 operates in HF mode when the EUSART CLK clock selected in CMU_EUSARTnCLKCTRL.CLKSEL is EM01GRPACLK or HFRCOEM23. EUSART0 operates in LF mode when the selected clock is LFXO or LFRCO. Baud rate generation differs between these two modes, and there are certain operational restrictions in LF mode discussed in this chapter. It is not generally useful to switch between modes on-the-fly in a single application.

Other EUSART instances operate only as a high-speed peripheral running from the EM01GRPCCLK selected using CMU_EM01GRPCCLKCTRL.CLKSEL.

21.3.3.1 Frame Format

The frame format used in asynchronous mode consists of a set of data bits in addition to bits for synchronization and optionally a parity bit for error checking. A frame starts with one start-bit (S), where the line is driven low for one bit-period. This signals the start of a frame, and is used for synchronization. Following the start bit are 7 to 9 data bits and an optional parity bit. Finally, a number of stop-bits, where the line is driven high that indicates the end of the frame. An example frame is shown in Figure 21.2. Frame Format.

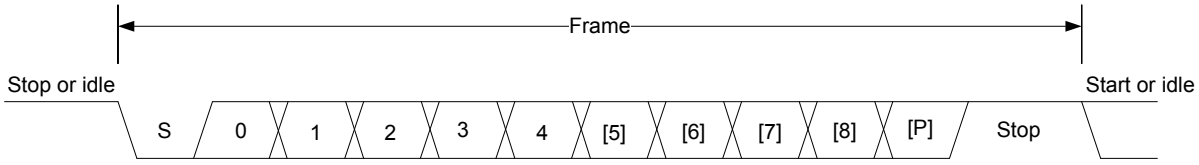


Figure 21.2. Frame Format

The number of data bits in a frame is set by DATABITS in EUSARTn_FRAMECFG, see Table 21.4. Data Bits, and the number of stop-bits is set by STOPBITS in EUSARTn_FRAMECFG, see Table 21.5. Stop Bits. Whether or not a parity bit should be included, and whether it should be even or odd is defined by PARITY, also in EUSARTn_FRAMECFG. For proper and reliable communication, all parties of a transfer must agree on the frame format prior to the start of the transfer.

Table 21.4. Data Bits

EUSARTn_FRAMECFG.DATABITS[3:0]	Number of Data bits
0001	7
0010	8 (Default)
0011	9

Table 21.5. Stop Bits

EUSARTn_FRAMECFG.STOPBITS[1:0]	Number of Stop bits
00	0.5 (HF clock operation only)
01	1 (Default)
10	1.5 (HF clock operation only)
11	2

The order in which the data bits are transmitted and received is defined by MSBF in EUSARTn_CFG0. When MSBF is cleared, data in a frame is sent and received with the least significant bit first. When it is set, the most significant bit comes first.

The frame format used by the transmitter can be inverted by setting TXINV in EUSARTn_CFG0, and the format expected by the receiver can be inverted by setting RXINV in EUSARTn_CFG0. These bits affect the entire frame, not only the data bits. An inverted frame has a low idle state, a high start-bit, inverted data and parity bits, and low stop-bits.

21.3.3.2 Parity Bit Calculation and Handling

When parity bit is enabled, hardware automatically calculates and inserts a parity bit into outgoing frames, and verifies the received parity bit in incoming frames. The possible parity modes are defined in Table 21.6. Parity Bits. When even parity is chosen, a parity bit is inserted to make the number of high bits (data + parity) even. If odd parity is chosen, the parity bit makes the total number of high bits odd. When parity bit is disabled, which is the default configuration, the parity bit is omitted.

Table 21.6. Parity Bits

EUSARTn_FRAMECFG.PARITY[1:0]	Description
00	No parity bit (Default)
01	Reserved
10	Even parity
11	Odd parity

21.3.3.3 Clock Generation

The EUSART clock defines the transmission and reception data rate. The baud rate is given by Figure 21.3. EUSART Baud Rate.

$$br = f_{EUSARTn}/(oversample \times (1 + EUSARTn_CLKDIV/256))$$

Figure 21.3. EUSART Baud Rate

where $f_{EUSARTn}$ is the peripheral clock frequency and oversample is the oversampling rate as defined by OVS in EUSARTn_CFG0, see Table 21.7. Oversampling (EUSARTn_CFG0.OVS). Note that different instances of the EUSART inside a device may use different peripheral clocks. The peripheral clock may be generically referred to in this chapter as clk_per.

Note:

Please note that when EUSARTn_CFG0.OVS is set to DISABLE, the peripheral clock frequency must be at least three times higher than the chosen baud rate. This condition is given in Figure 21.4. Requirement for EUSARTn_CFG0.OVS = DISABLE.

$$f_{EUSARTn} / br >= 3.0$$

Figure 21.4. Requirement for EUSARTn_CFG0.OVS = DISABLE

Table 21.7. Oversampling (EUSARTn_CFG0.OVS)

OVS[2:0]	Oversample
000	16 (HF clock operation only)
001	8 (HF clock operation only)
010	6 (HF clock operation only)
011	4 (HF clock operation only)
100	1 (OVS disabled - LF clock operation only)

Note: Please note that EUSARTn_CFG0.OVS must not be set to DISABLE when one of the high frequency clocks is selected as EUSARTn peripheral clock source. When one of the low frequency clocks (LFXO/LFRCO) is selected as EUSARTn peripheral clock source, EUSARTn_CFG0.OVS must be set to DISABLE.

The EUSART has a fractional clock divider to allow the EUSART clock to be controlled more accurately than what is possible with a standard integral divider. The clock divider used in the EUSART is a 20-bit value, with a 15-bit integral part and an 5-bit fractional part. The fractional part is configured in the lower 5 bits of DIV in EUSARTn_CLKDIV. Fractional clock division is implemented by distributing the selected fraction over thirty two baud periods. The fractional part of the divider tells how many of these periods should be extended by one peripheral clock cycle.

Given a desired baud rate *brdesired*, the clock divider EUSARTn_CLKDIV can be calculated by using Figure 21.5. EUSART Desired Baud Rate:

$$EUSARTn_CLKDIV = 256 \times (f_{EUSARTn}/(oversample \times brdesired) - 1)$$

Figure 21.5. EUSART Desired Baud Rate

Table 21.8. EUSART Baud Rates @ 4 MHz Peripheral Clock with 20 Bit CLKDIV shows a set of desired baud rates and how accurately the EUSART is able to generate these baud rates when running at a 4 MHz peripheral clock, using 16x or 8x oversampling.

Table 21.8. EUSART Baud Rates @ 4 MHz Peripheral Clock with 20 Bit CLKDIV

Desired baud rate [baud/s]	EUSARTn_CFG0.OVS =000			EUSARTn_CFG0.OVS =001		
	EUSARTn_CLKDIV/256 (to 32nd position)	Actual baud rate [baud/s]	Error %	EUSARTn_CLKDIV/256 (to 32nd position)	Actual baud rate [baud/s]	Error[1] [%]
600	415,6563	600,015	0,003	832,3438	599,9925	-0,001

Desired baud rate [baud/s]	EUSARTn_CFG0.OVS =000			EUSARTn_CFG0.OVS =001		
	EUSARTn_CLKDIV/256 (to 32nd position)	Actual baud rate [baud/s]	Error %	EUSARTn_CLKDIV/256 (to 32nd position)	Actual baud rate [baud/s]	Error[1] [%]
1200	207,3438	1199,94	-0,005	415,6563	1200,03	0,003
2400	103,1563	2400,24	0,010	207,3438	2399,88	-0,005
4800	51,09375	4799,04	-0,020	103,1563	4800,48	0,010
9600	25,03125	9603,842	0,040	51,09375	9598,08	-0,020
14400	16,375	14388,49	-0,080	33,71875	14401,44	0,010
19200	12,03125	19184,65	-0,080	25,03125	19207,68	0,040
28800	7,6875	28776,98	-0,080	16,375	28776,98	-0,080
38400	5,5	38461,54	0,160	12,03125	38369,3	-0,080
57600	3,34375	57553,96	-0,080	7,6875	57553,96	-0,080
76800	2,25	76923,08	0,160	5,5	76923,08	0,160
115200	1,15625	115942	0,644	3,34375	115107,9	-0,080
230400	0,09375	228571,4	-0,794	1,15625	231884,1	0,644

1 In addition to error introduced by oscillator frequency variation

Table 21.9. EUSART0 Baud Rates in LF mode @ 32.768 kHz Peripheral Clock with 20 bit CLKDIV shows a set of desired baud rates and how accurately EUSART0 is able to generate these baud rates when running from a 32.768 kHz peripheral clock in LF mode.

Table 21.9. EUSART0 Baud Rates in LF mode @ 32.768 kHz Peripheral Clock with 20 bit CLKDIV

Desired baud rate [baud/s]	EUSARTn_CLKDIV/256	Actual baud rate [baud/s]	Error[1] [%]
300	108,21875	300,0217	0.01
600	53,625	599,8719	-0.02
1200	26,3125	1199,744	-0.02
2400	12,65625	2399,487	-0.02
4800	5,8125	4809,982	0.21
9600	2,40625	9619,963	0.21

1 In addition to error introduced by oscillator frequency variation

21.3.3.4 Auto Baud Detection (HF clock only)

Setting AUTOBAUDEN in EUSARTn_CFG0 uses the first frame received to automatically set the baud rate provided that it contains 0x55 (IrDA uses 0x00) and is sent out as LSB first and there is no break in the frame. The receiver will measure the number of local clock cycles between the beginning of the START bit and the beginning of the 8th data bit. The DIV field in EUSARTn_CLKDIV will be overwritten with the new value. The OVS in EUSARTn_CFG0 and the +1 count of the Baud Rate equation are already factored into the result that gets written into the DIV field. To restart autobaud detection, clear AUTOBAUDEN and set it high again. Since the auto baud rate detection is done over 8 baud times, only the upper 3 bits of the fractional part of the clock divider are populated.

Auto baud detection has associated status bit EUSARTn_STATUS.AUTOBAUDDONE and interrupt flag EUSARTn_IF.AUTOBAUDDONE. Both the status and the interrupt flag get set after auto baud detection is complete and DIV field in EUSARTn_CLKDIV is overwritten with the new value.

Note:

- If autobaud detection is enabled, software must wait for autobaud detection to complete before transmitting any data.
- Autobaud should be used only during times when it is known that the transmitter will be sending the required data word.
- Autobaud detection is not available when operating with LF clock source.
- For autobaud to work with IrDA, there should be odd parity or no parity in the received data frame.

21.3.3.5 Data Transmission

Asynchronous data transmission is initiated by writing data to the transmit FIFO using one of the methods described in 21.3.3.5.1 Transmit FIFO Operation. When the transmission shift register is empty and if the transmitter is enabled, a frame from the transmit FIFO is loaded into the shift register and transmission begins. Note that the frame loading in to the shift register can also be blocked if CTSEN is set in EUSARTn_CFG0 but the CTS input is inactive. When the frame has been transmitted, a new frame is loaded into the shift register if available, and transmission continues. If the transmit FIFO is empty, the transmitter goes to an idle state, waiting for a new frame to become available.

Transmission is enabled through the command register EUSARTn_CMD by setting TXEN, and disabled by setting TXDIS in the same command register. When the transmitter is disabled using TXDIS, any ongoing transmission is aborted, and any frame currently being transmitted is discarded. When disabled, the TX output goes to an idle state, which by default is a high value. Whether or not the transmitter is enabled at a given time can be read from TXENS in EUSARTn_STATUS.

When the EUSART transmitter is enabled and there is no data in the transmit shift register or transmit FIFO, the TXC status flag in EUSARTn_STATUS and the TXC interrupt flag in EUSARTn_IF are set, signaling that the transmission is complete. The TXC status flag is cleared when a new frame becomes available for transmission, but the TXC interrupt flag must be cleared by software.

Note: (1) Condition for TX to send data out: TX is enabled and there is data available in the TX FIFO and CTS is either not enabled, or if it is enabled, then it is active. (2) If TX output is tri-stated using TXDIS command or TXDISAT setting, then the TX module will still send out data (emptying the FIFO) if condition in point 1 is satisfied, even though the pad is tristated. Restriction: User should not set TXEN when the output is tristated.

21.3.3.5.1 Transmit FIFO Operation

The transmit FIFO is a 32 deep FIFO. A frame can be loaded into the FIFO by writing to EUSARTn_TXDATA. Using EUSARTn_TXDATA allows 9 bits to be written to the FIFO, as well as a set of control bits regarding the transmission of the written frame. Every frame in the FIFO is stored with 9 data bits and additional transmission control bits. A frame is loaded from the FIFO in to the shift register if the transmitter is enabled.

Figure 21.6. Transmit FIFO Operation shows the basics of the transmit FIFO.

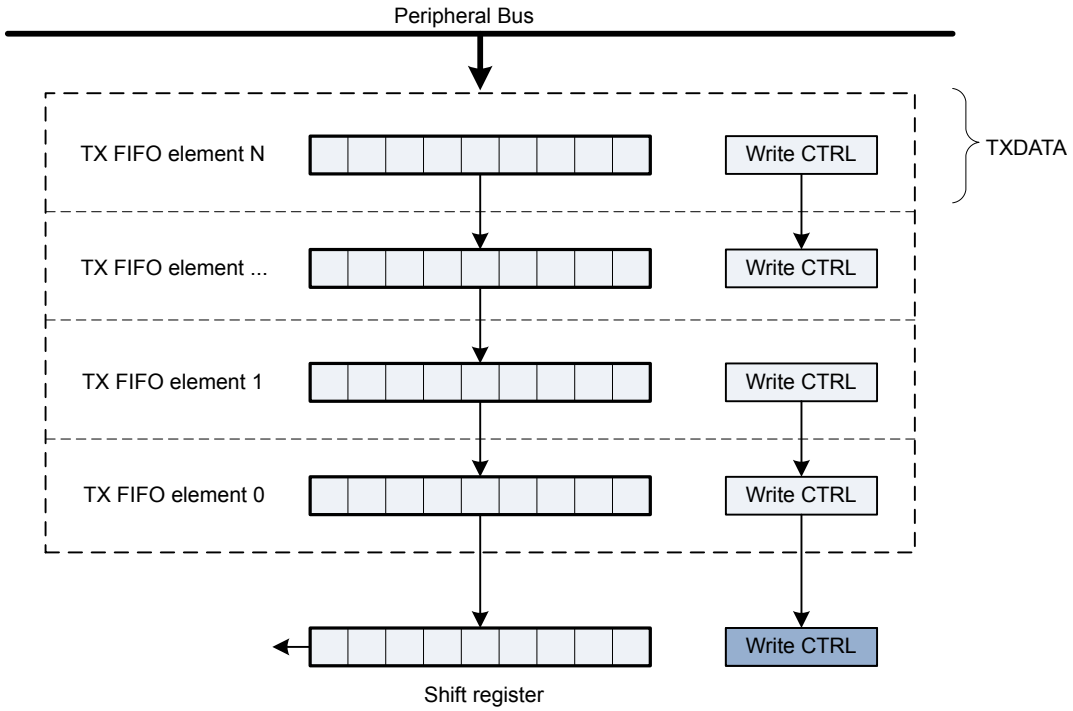


Figure 21.6. Transmit FIFO Operation

In addition to the interrupt flag TXC in EUSARTn_IF and status flag TXC in EUSARTn_STATUS which are set when the transmission is complete, TXFL in EUSARTn_STATUS and the TXFL interrupt flag in EUSARTn_IF are used to indicate the level of the transmit FIFO. TXFIW in EUSARTn_CFG1 controls the level at which these bits are set. The user can choose a transmit FIFO watermark level in TXFIW field of EUSARTn_CFG1 register in order to program when the TXFL interrupt should be triggered.

There is a TXIDLE status bit in EUSARTn_STATUS to provide an indication of when the transmitter is idle. The count of TX valid FIFO entries is called TXFCNT in EUSARTn_STATUS.

The TXCANDTXIDLE bit in EUSARTn_STATUS is set to 1 when both TXC and TXIDLE are 1.

The transmit FIFO can be cleared by setting CLEARTX in EUSARTn_CMD. Since this is an Async FIFO, the user needs to first issue the CLEARTX command and then wait on the CLEARTXBUSY status flag until it goes low. EUSART must be enabled (EUSARTn_EN should be set) for the flush to work. EUSART should not be transmitting when CLEARTX command gets issued (can be achieved by disabling the trasmitter). Any frame present in the transmit shift register currently being transmitted will not be aborted due to the flush. Transmission of this frame will be completed. Note that the transmit shift register is never used to store a transmit frame, i.e., if the transmitter is not enabled then the data stays in the transmit FIFO until the transmitter gets enabled. Whenever a frame is loaded in to the transmit shift register, the transmission starts immediately.

Transmit FIFO Status Flags

TX FIFO has two associated status flags: TXFL (set when there is space for data in the TX FIFO, depends on the setting in the TXFIW in CFG1 register) and TXFCNT (count of number of TX frames in the FIFO). These status flags remain set as long as the underlying condition is true, even if the EUSART is disabled (i.e., EUSARTn_EN is 0). It is possible to write to the TX FIFO while EUSART is disabled, this will impact the two flags mentioned above.

Transmit FIFO Interrupt Flags

TX FIFO has two associated interrupt flags: TXFL and TXOF. TXFL is set when space becomes available in the FIFO, depends on TXFIW in CFG1 register. Reset value for TXFIW is such that TXFL gets set as soon the chip comes out of reset (so the interrupt should be cleared once TXFIW is updated). TXFL remains set as long as the underlying condition is true even if the EUSART is disabled. This means that if a software clear is done for TXFL, then the interrupt will get set again after the clear if the underlying condition is still true (this will happen even if the EUSART is disabled). Writing more data to the FIFO or disabling the TXFL will make the TXFL go away (even if the EUSART is disabled). When writing more frames to the transmit FIFO than there is free space for, the TXOF interrupt flag in EUSARTn_IF will be set, indicating the overflow. The data already in the transmit FIFO is preserved in this case, and no data is written. Note that TXOF can also trigger if DMA tries to write to the FIFO while a TX Flush operation is going on (Flush is activated by setting CMD.CLEARTX). No data will be written to the FIFO in this case. TXOF triggers only once every time an overflow occurs.

Note: In LF clock mode (EUSART0 only) , the TXFL interrupt flag and the TXFL wakeup flag differ slightly in their behavior. IF.TXFL is always set out of reset since there is space available in the FIFO. However, the TXFL wakeup flag is only set after a FIFO read happens and the space that becomes available in the FIFO is the same as programmed in TXFIW in EUSARTn_CFG1.

21.3.3.5.2 Frame Transmission Control

The transmission control bits, which can be written using EUSARTn_TXDATA, affect the transmission of the written frame. The following options are available:

- Generate break: By setting TXBREAK, the output will be held low during the stop-bit period to generate a framing error. A receiver that supports break detection detects this state, allowing it to be used e.g. for framing of larger data packets. The line is driven high before the next frame is transmitted so the next start condition can be identified correctly by the recipient. Continuous breaks lasting longer than a EUSART frame are thus not supported. GPIO can be used for this.
- Disable transmitter after transmission: If TXDISAT is set, the transmitter is disabled after the frame has been fully transmitted.
- Enable receiver after transmission: If RXENAT is set, the receiver is enabled after the frame has been fully transmitted. It is enabled in time to detect a start-bit directly after the last stop-bit has been transmitted.
- Unblock receiver after transmission: If UBRXAT is set, the receiver is unblocked and RXBLOCK is cleared after the frame has been fully transmitted.
- Tristate transmitter after transmission: If TXTRIAT is set, TXTRI is set after the frame has been fully transmitted, tristating the transmitter output. Note that if there are more frames in the TX FIFO after the tristating has happened and the transmitter is enabled, then the transmitter will try to send them out but since the output is tristated, nothing will show up at the transmitter output. The FIFO however will get emptied because of the transmitter attempting to send these frames out. If the target is to automatically tristate the TX line whenever the transmitter is idle, then that can be done by setting AUTOTRI in EUSARTn_CFG0. If AUTOTRI is set TXTRI status flag is always read as 0.

21.3.3.5.3 TX Status Flags

The following status flags should be used with care keeping in mind the external sources that can impact these flags as well as the synchronization delay when the status signal crosses over from the EUSART Core clock domain to the APB clock domain. FIFO related status flags were already discussed in 21.3.3.5.1 Transmit FIFO Operation, the remaining flags are mentioned below:

- TXENS: Enable sources: (1) TXEN command from software, (2) PRS trigger (when TXTEN=1). Disable Sources: (1) TXDIS command from software, (2) PERR/FERR (when ERRSTX=1), (3) Software when TXDISAT is set to 1 for a frame.
- TXTRI: Enable sources: (1) TXTRIEN command from software, (2) Software when TXTRIAT is set to 1 for a frame. Disable sources: (1) TXTRIDIS command from software.
- TXIDLE: Set whenever the TX module is idle.
- TXC: Set whenever the TX module goes to idle and both the TX FIFO and the TX shift register are empty. Cleared when either TX FIFO or the TX shift register has data.
- TXCANDTXIDLE: Set when both TXC and TXIDLE are set.

21.3.3.5.4 Transmission Delay

By configuring TXDELAY in EUSARTn_CFG1, the transmitter can be forced to wait a number of bit-periods from when it is ready to transmit data, to when it actually transmits the data. This delay is only applied to the first frame transmitted after the transmitter has been idle. When transmitting frames back-to-back the delay is not introduced between the transmitted frames.

This is useful on half duplex buses, because the receiver always returns received frames to software during the first stop-bit. The bus may still be driven for up to 3 bit periods, depending on the current frame format. Using the transmission delay, a transmission can be started when a frame is received, and it is possible to make sure that the transmitter does not begin driving the output before the frame on the bus is completely transmitted.

21.3.3.6 Data Reception

Data reception is enabled by setting RXEN in EUSARTn_CMD. When the receiver is enabled, it actively samples the input looking for a transition from high to low indicating the start baud of a new frame. When a start baud is found, reception of the new frame begins. When the frame has been received, it is pushed into the receive FIFO, making the shift register ready for another frame of data, and the receiver starts looking for another start baud. If a frame is received while the receive FIFO is full, the received frame is discarded and the RXOF interrupt flag in EUSARTn_IF is set to indicate the FIFO overflow.

The receiver can be disabled by setting the command bit RXDIS in EUSARTn_CMD. Any frame currently being received, when the receiver is disabled, is discarded. Whether or not the receiver is enabled at a given time can be read out from RXENS in EUSARTn_STATUS.

21.3.3.6.1 Receive FIFO Operation

The receive-FIFO is a 32 deep FIFO. Data can be read from the receive FIFO via EUSARTn_RXDATA. EUSARTn_RXDATA gives access to the received frame. This register also contains parity error and framing error information of the received frame. When a frame is read from the receive FIFO using EUSARTn_RXDATA, the frame is pulled out of the FIFO, making room for a new frame. If an attempt is done to read more frames from the FIFO than what is available, the RXUF interrupt flag in EUSARTn_IF is set to signal the underflow, and the data read from the FIFO is undefined.

Frames can be read from the receive FIFO without removing the data by using EUSARTn_RXDATAP. EUSARTn_RXDATAP gives access to the first frame in the FIFO with status bits. The data read from this register when the receive FIFO is empty is undefined. No underflow interrupt is generated by a read using this register, i.e. RXUF in EUSARTn_IF is never set as a result of reading from EUSARTn_RXDATAP.

The basic operation of the receive FIFO is shown in Figure 21.7. EUSART Receive FIFO Operation.

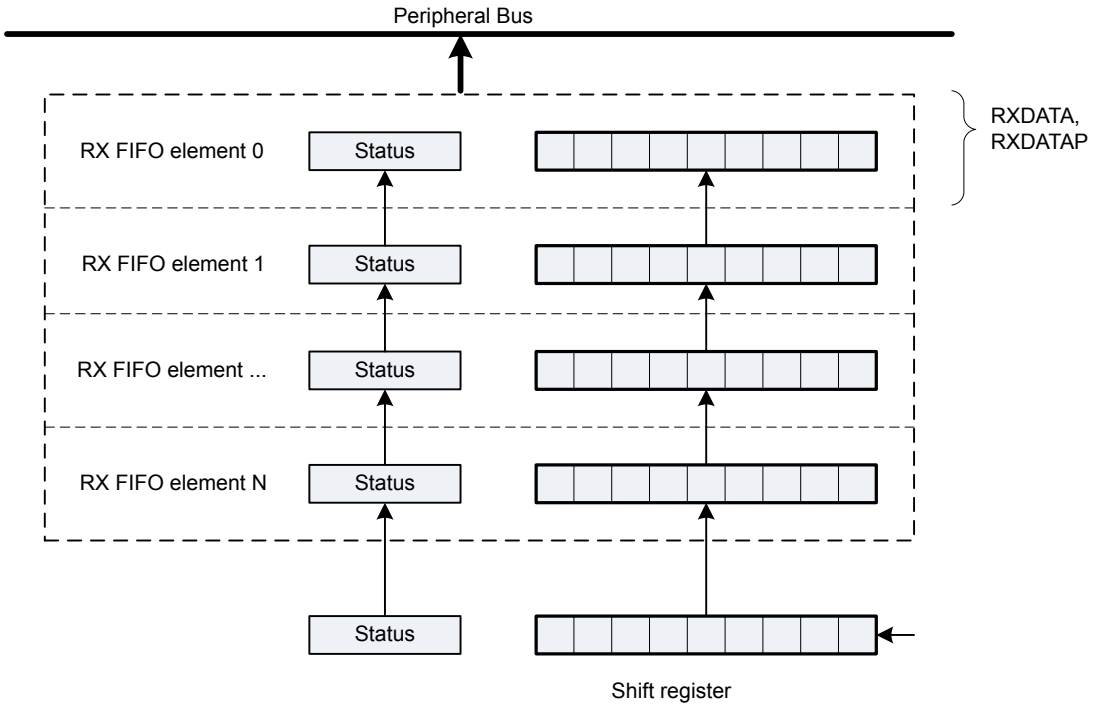


Figure 21.7. EUSART Receive FIFO Operation

The RXFCNT field in the EUSARTn_STATUS register indicates the number of valid entries in the RXFIFO. The Receive FIFO can be cleared by setting CLEARRX in EUSARTn_CMD. Since this is an asynchronous FIFO, the user needs to first issue the CLEARRX command and then wait on the CLEARRXBUSY status flag until it goes low. EUSART must be enabled (EUSARTn_EN should be set) for the flush to work. EUSART should not be receiving when CLEARRX command gets issued which can be achieved by disabling/blocking the receiver (reciever blocking feature not supported in DALI mode).

Receive FIFO Status Flags

Receive FIFO has two associated status flags: RXFL (set when number of available frames in the receive FIFO is at least number of frames set by RXFIW in CFG1 register) and RXFULL (set when receive FIFO is full). These status flags remain set as long as the underlying condition is true, even if the EUSART is disabled (i.e., EUSARTn_EN is 0). It is possible to read from the receive FIFO while EUSART is disabled, this will impact the two status flags mentioned above. The status flags RXFL and RXFULL are automatically cleared by hardware when their condition is no longer true.

Receive FIFO Interrupt Flags

Receive FIFO has four associated interrupt flags: RXFL, RXFULL, RXOF and RXUF. RXFL is set when number of available frames in the receive FIFO is at least number of frames set by RXFIW in CFG1 register. RXFULL is set when receive FIFO is full. Both RXFL and RXFULL remains set as long as the underlying condition is true even if EUSART is disabled. This means that if a software clear is done for RXFL / RXFULL while the undelying condition of respective interrupt is still true, the corresponding interrupt will get set again after the clear (this will happen even if EUSART is disabled). Reading data from the FIFO or disabling the RXFL / RXFULL will block

the respective interrupt to get set after a software clear (even if EUSART is disabled). RXOF is set when a new frame is received while the receive FIFO is full, indicating overflow. The new frame is discarded. RXOF triggers only once every time an overflow occurs. RXUF is set when an attempt (via RXDATA or DMA) is done to read more frames from the receive FIFO than what is available, indicating underflow. The data read from the FIFO is undefined. RXUF triggers every time an underflow occurs even if EUSART is disabled.

21.3.3.6.2 Blocking Incoming Data

When using hardware frame recognition, as detailed in 21.3.3.6.9 Multi-Processor Mode and 21.3.3.8 Collision Detection, it is necessary to be able to let the receiver sample incoming frames without loading them into the receive FIFO. This is accomplished by blocking incoming data.

Incoming data is blocked as long as RXBLOCK in EUSARTn_STATUS is set. When blocked, frames received by the receiver will not be loaded into the receive FIFO, and software is not notified by the RXFL flag in EUSARTn_STATUS or the RXFL interrupt flag in EUSARTn_IF at their arrival. For data to be loaded into the receive FIFO, RXBLOCK must be cleared in the instant a frame is fully received by the receiver. RXBLOCK is set by setting RXBLOCKEN in EUSARTn_CMD and disabled by setting RXBLOCKDIS also in EUSARTn_CMD. There is one exception where data is loaded into the receive FIFO even when RXBLOCK is set. This is when an address frame is received while operating in multi-processor mode. See 21.3.3.6.9 Multi-Processor Mode for more information.

Frames received containing framing or parity errors will not result in the FERR and PERR interrupt flags in EUSARTn_IF being set while RXBLOCK in EUSARTn_STATUS is set. Hardware recognition is not applied to these erroneous frames, and they are silently discarded.

The overflow interrupt flag RXOF in EUSARTn_IF will also not be set while RXBLOCK in EUSARTn_STATUS is set.

21.3.3.6.3 Data Sampling and Filtering

The receiver can sample incoming signal at a rate 16, 8, 6 or 4 times higher than the given baud rate, depending on the oversampling mode given by OVS in EUSARTn_CFG0. Lower oversampling rates make higher baud rates possible, but give less room for errors.

When a high-to-low transition is registered on the input while the receiver is idle, this is recognized as a start-bit, and the baud rate generator is synchronized with the incoming frame.

For oversampling modes 16, 8 and 6, every bit in the incoming frame is sampled three times to gain a level of noise immunity. These samples are aimed at the middle of the bit-periods, as visualized in Figure 21.8. EUSART Sampling of Start and Data Bits. With OVS=0 in EUSARTn_CFG0, the start and data bits are thus sampled at locations 8, 9 and 10 in the figure, locations 4, 5 and 6 for OVS=1 and locations 3, 4, and 5 for OVS=2. The value of a sampled bit is determined by majority vote. If two or more of the three bit-samples are high, the resulting bit value is high. If the majority is low, the resulting bit value is low.

Majority vote is used for all oversampling modes except 4X oversampling and when oversampling is disabled. In 4X oversampling mode, a single sample is taken at position 3 as shown in Figure 21.8. EUSART Sampling of Start and Data Bits.

When oversampling is disabled i.e. OVS = DISABLE, there is only one available location for sampling the start and data bits and so majority vote is not used.

Note: When operating in HF clock mode, oversampling must be set to 4, 6, 8, or 16x. When operating in LF clock mode, oversampling must be disabled.

Software can disable majority vote by setting MVDIS in EUSARTn_CFG0. When majority vote is disabled by software, a single sample is taken at location 9 in the figure for OVS=0, location 5 for OVS=1 and location 4 for OVS=2.

If the value of the start bit is found to be high, the reception of the frame is aborted, filtering out false start bits possibly generated by noise on the input.

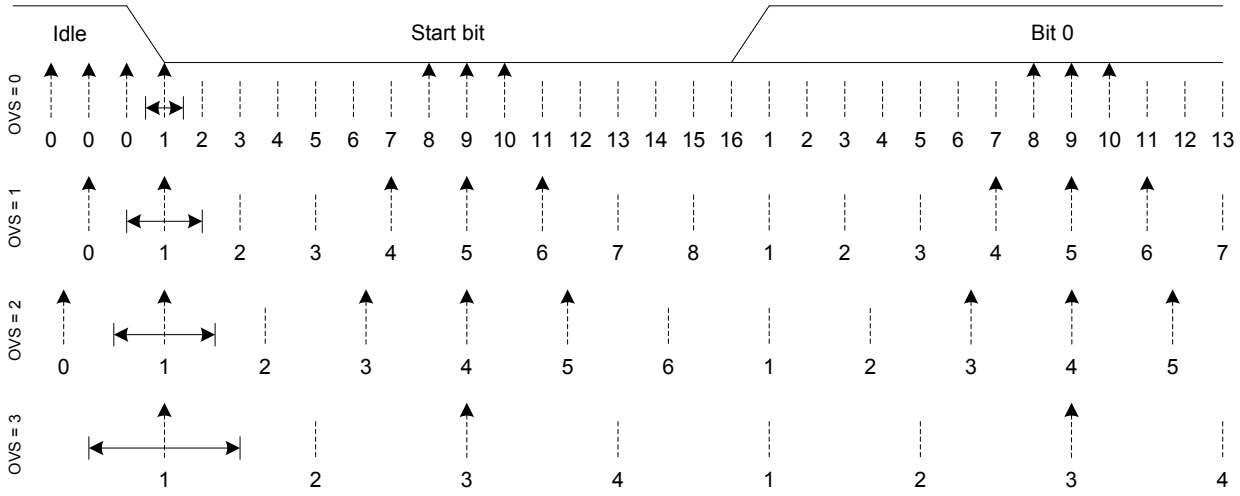


Figure 21.8. EUSART Sampling of Start and Data Bits

If the baud rate of the transmitter and receiver differ, the location each bit is sampled will be shifted towards the previous or next bit in the frame. This is acceptable for small errors in the baud rate, but for larger errors, it will result in transmission errors.

When the number of stop bits is 1 or more, stop bits are sampled like the start and data bits as seen in Figure 21.9. EUSART Sampling of Stop Bits when Number of Stop Bits are 1 or More. When a stop bit has been detected, EUSARTn is ready for a new start bit.

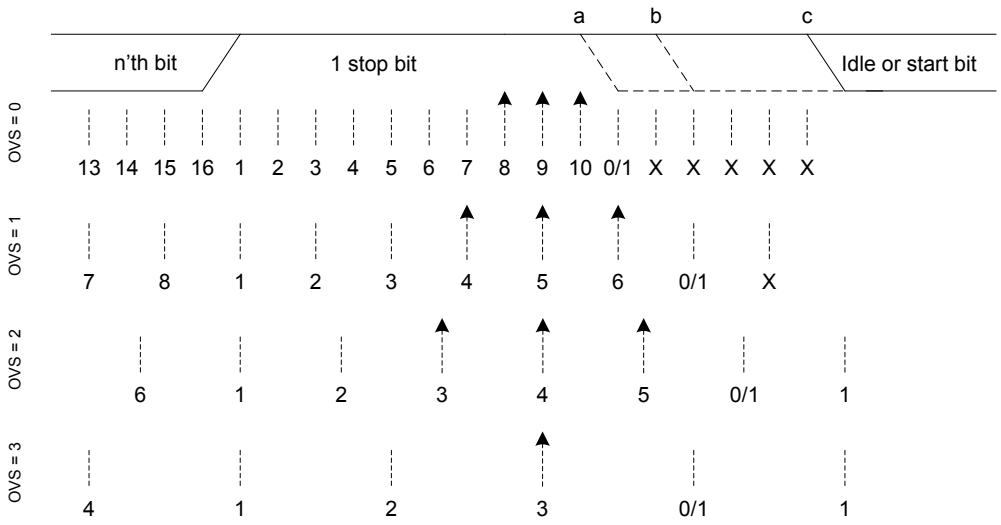


Figure 21.9. EUSART Sampling of Stop Bits when Number of Stop Bits are 1 or More

When working with stop bit lengths of half a baud period, the above sampling scheme no longer suffices. In this case, the stop-bit is not sampled, and no framing error is generated in the receiver if the stop-bit is not generated. However, the line must still be driven high before the next start bit for the EUSART to successfully identify the start bit.

21.3.3.6.4 RX Status Flags

The following status flags should be used with care keeping in mind the external sources that can impact these flags as well as the synchronization delay when the status signal crosses over from the EUSART Core clock domain to the APB clock domain. FIFO related status flags were already discussed in 21.3.3.6.1 Receive FIFO Operation, the remaining flags are mentioned below:

- RXENS: set when RX is enabled. Enable sources: (1) RXEN command from software, (2) PRS trigger (when RXTEN=1), (3) TX (when RXENAT=1). Disable Sources: (1) RXDIS command from software, (2) Parity / framing error (when ERRSRX=1).
- RXBLOCK: set when RX is blocked which means RX keeps receiving frames but does not push received frame to receive FIFO. Enable sources: (1) RXBLOCKEN command from software. Disable Sources: (1) RXBLOCKDIS command from software, (2) STARTFRAME match (when SFUBRX=1), (3) TX (when UBRXAT=1).
- RXIDLE: Set when RX module completes pushing the last received frame to receive FIFO (if receive FIFO has space) and is not receiving any new frame or when RX gets disabled.
- AUTOBAUDDONE: Set when auto baud detection is complete and DIV field in EUSARTn_CLKDIV is overwritten with the detected value.

21.3.3.6.5 Parity Error

When parity bits are enabled, a parity check is automatically performed on incoming frames. When a parity error is detected in an incoming frame, the data parity error bit PERR in the frame is set, as well as the interrupt flag PERR in EUSARTn_IF. Frames with parity errors are loaded into the receive FIFO like regular frames.

PERR can be accessed by reading the frame from the receive FIFO using the EUSARTn_RXDATA, or EUSARTn_RXDATAP registers.

If ERRSTX in EUSARTn_CFG0 is set, the transmitter is disabled on received parity errors. If ERRSRX in EUSARTn_CFG0 is set, the receiver is disabled on parity errors.

21.3.3.6.6 Framing Error and Break Detection

A framing error is the result of an asynchronous frame where the stop bit was sampled to a value of 0. This can be the result of noise and baud rate errors, but can also be the result of a break generated by the transmitter on purpose.

When a framing error is detected in an incoming frame, the framing error bit FERR in the frame is set. The interrupt flag FERR in EUSARTn_IF is also set. Frames with framing errors are loaded into the receive FIFO like regular frames.

If ERRSTX in EUSARTn_CFG0 is set, the transmitter is disabled on received framing errors. If ERRSRX in EUSARTn_CFG0 is set, the receiver is disabled on framing errors.

21.3.3.6.7 Programmable Start Frame

The EUSART can be configured to start receiving data when a special start frame is detected on the input. This can be useful when operating in low energy modes, allowing other devices to gain the attention of the EUSART by transmitting a given frame.

When SFUBRX in EUSARTn_CFG1 is set, an incoming frame matching the frame defined in EUSARTn_STARTFRAMECFG will result in RXBLOCK in EUSARTn_STATUS being cleared. This can be used to enable reception when a specified start frame is detected. If the receiver is enabled and blocked, i.e. RXENS and RXBLOCK in EUSARTn_STATUS are set, the receiver will receive all incoming frames, but unless an incoming frame is a start frame it will be discarded and not loaded into the receive FIFO. When a start frame is detected, the block is cleared, and frames received from that point, including the start frame, are loaded into the receive FIFO.

An incoming start frame results in the STARTFIF interrupt flag in EUSARTn_IF being set, regardless of the value of SFUBRX in EUSARTn_CFG1. This allows an interrupt to be made when the start frame is detected. The interrupt will be set even if the receiver is blocked i.e. EUSARTn_STATUS.RXBLOCK = 1.

Note: The receiver must be enabled for start frames to be detected. Please note that, if another UART device sends a start frame but a parity and/or framing error occurs during the receiption, the received frame is not detected as a start frame.

21.3.3.6.8 Programmable Signal Frame

As well as the configurable start frame, a special signal frame can be specified. When a frame matching the frame defined in EUSARTn_SIGFRAMECFG is detected by the receiver, the SIGF interrupt flag in EUSARTn_IF is set. As like start frame detection, the interrupt will be set even if the receiver is blocked i.e. EUSARTn_STATUS.RXBLOCK = 1.

One use of the programmable signal frame is to signal the end of a multi-frame message transmitted to the EUSART. An interrupt will then be triggered when the packet has been completely received, allowing software to process it. Used in conjunction with the programmable start frame and DMA, this makes it possible for the EUSART to automatically begin the reception of a packet on a specified start frame, load the entire packet into memory, and give an interrupt when reception of a packet has completed. When one of the low frequency oscillators (LFXO, LFRCO) is used as the EUSART0 peripheral clock source, the device can thus wait for data packets in a low energy state, and only wake when a packet has been completely received.

Note: The receiver must be enabled for a signal frame to be detected. If a parity and/or framing error occurs during the reception of a signal frame, the received frame is not detected as a signal frame.

21.3.3.6.9 Multi-Processor Mode

To simplify communication between multiple processors, the EUSART supports a special multi-processor mode. In this mode the 9th data bit in each frame is used to indicate whether the content of the remaining 8 bits is data or an address.

When multi-processor mode is enabled, an incoming 9-bit frame with the 9th bit equal to the value of MPAB in EUSARTn_CFG0 is identified as an address frame. When an address frame is detected, the MPAF interrupt flag in EUSARTn_IF is set, and the address frame is loaded into the receive register. This happens regardless of the value of RXBLOCK in EUSARTn_STATUS.

Multi-processor mode is enabled by setting MPM in EUSARTn_CFG0, and the value of the 9th bit in address frames can be configured in EUSARTn_CFG0.MPAB. Note that the receiver must be enabled for address frames to be detected. The receiver can be blocked however, preventing data from being loaded into the receive FIFO while looking for address frames.

The following section explains basic usage of the multi-processor mode:

EUSART Multi-processor Mode Example

1. All devices on the bus enable multi-processor mode and enable and block the receiver. They will now not receive data unless it is an address frame. MPAB in EUSARTn_CFG0 is set to desired value to identify frames with the 9th bit value equal to MPAB as address frames.
2. A transmitting device sends a frame containing the address of a different device and with the 9th bit set to value of MPAB.
3. All devices receive the address frame and get an interrupt. They can read the address from the receive FIFO. The selected device unblocks the receiver to start receiving data from the transmitter.
4. The transmitter sends data with the 9th bit set to opposite value of MPAB.
5. Only the addressed device with RX enabled and unblocked receives the data. When transmission is complete, the device blocks the receiver and waits for a new address frame.

When a device has received an address frame and wants to receive the following data, it must make sure the receiver is unblocked before the next frame has been completely received in order to prevent data loss. One option is to set SFUBRX in EUSARTn_CFG1 and set the address frame as start frame in EUSARTn_STARTFRAMECFG. This will handle automatic unblocking of RX when address frame is received.

21.3.3.6.10 RX Timeout

A RX timeout function can be enabled by setting EUSART_CFG1.RXTIMEOUT to desired value. When enabled, a timer gets started after every successful frame reception. If timeout occurs before the next RX start bit is received, EUSART_IF.RXTO gets set which can be used to wake-up the system to handle received data. This is shown in Figure 21.10. RX Timeout If the next RX start bit is received before timeout occurs, no interrupt gets generated, the timer is reset and will only be started again after the on-going frame reception is complete.

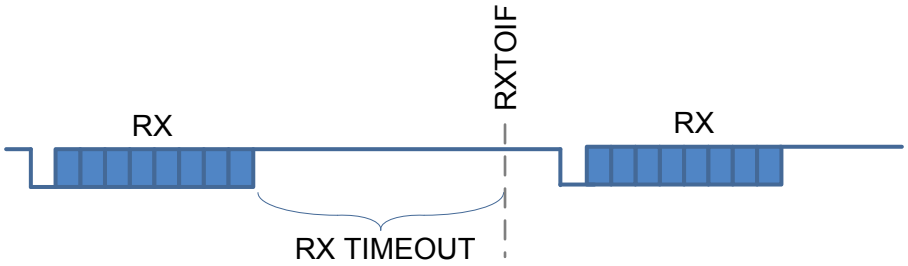


Figure 21.10. RX Timeout

Please note that the timer does not get started in following scnearios:

- If auto baud rate detection is enabled and auto baud rate has not been found, the timer does not get started after the first frame which is used to detect the baud rate automatically.
- If a frame is received while RX is blocked, the timer does not get started.
- If EUSART_CFG0.SKIPPERRF is set to '1' and a frame is received with a parity error, the timer does not get started.

Please also note that the UART RX input line must be in idle state after frame reception for the timer to start.

Note: When EUSARTn_FRAMECFG.STOPBITS is set to a fractional value i.e. 'HALF' or 'ONEANDHALF', the fractional value is rounded to nearest value and so RX timeout is extended by 0.5 baud width per frame.

21.3.3.7 Local Loopback

The EUSARTn receiver samples RX by default, and the transmitter drives TX by default. This is not the only option however. When LOOPBK in EUSARTn_CFG0 is set, the RX pin is connected to the TX pin as shown in Figure 21.11. EUSART Local Loopback. This is useful for debugging, as the EUSARTn can receive the data it transmits, but it is also used to allow the EUSARTn to read and write to the same pin, which is required for some half duplex communication modes. In this mode, the TX pin must be enabled as an output in the GPIO.

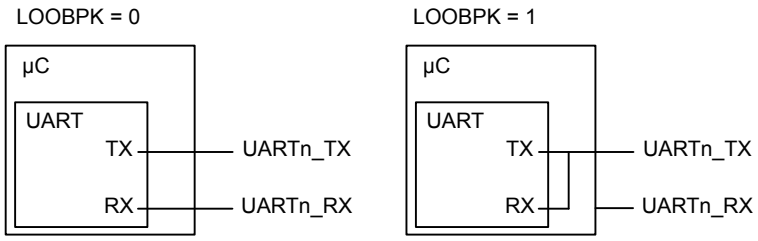


Figure 21.11. EUSART Local Loopback

21.3.3.8 Collision Detection

EUSARTn supports a basic form of collision detection. When the receiver is connected to the output of the transmitter, either by using the LOOPBK bit in EUSARTn_CFG0 or through an external connection, this feature can be used to detect whether data transmitted on the bus by the EUSARTn did get corrupted by a simultaneous transmission by another device on the bus.

For collision detection to be enabled, CCEN in EUSARTn_CFG0 must be set, and the receiver enabled. The data sampled by the receiver is then continuously compared with the data output by the transmitter. If they differ, the CCF interrupt flag in EUSARTn_IF is set. The collision check includes all bits of the transmitted frames. The CCF interrupt flag is set once for each bit sampled by the receiver that differs from the bit output by the transmitter. When the transmitter output is disabled, i.e. the transmitter is tristated, collisions are not registered.

Note: Please note that collision detection supports only baudrates up to 1mbps.

21.3.3.9 Half Duplex Communication

When doing full duplex communication, two data links are provided, making it possible for data to be sent and received at the same time. In half duplex mode, data is only sent in one direction at a time. There are several possible half duplex setups, as described in the following sections.

21.3.3.9.1 Single Data-link

In this setup, the EUSART both receives and transmits data on the same pin. This is enabled by setting LOOPBK in EUSARTn_CFG0, which connects the receiver to the transmitter output. Because they are both connected to the same line, it is important that the EUSART transmitter does not drive the line when receiving data, as this would corrupt the data on the line.

When communicating over a single data-link, the transmitter must thus be tristated whenever not transmitting data. This is done by setting the command bit TXTRIEN in EUSARTn_CMD, which tristates the transmitter. Before transmitting data, the command bit TXTRIDIS, also in EUSARTn_CMD, must be set to enable transmitter output again. Whether or not the output is tristated at a given time can be read from TXTRI in EUSARTn_STATUS. If TXTRI is set when transmitting data, the data is shifted out of the shift register, but is not put out on TX line.

When operating a half duplex data bus, it is common to have a main bus controller, which first transmits a request to one of the secondary devices on the bus, then receives a reply. In this case, the frame transmission control bits, which can be set by writing to EUSARTn_TXDATA, can be used to make the EUSART automatically disable transmission, tristate the transmitter and enable reception when the request has been transmitted, making it ready to receive a response from the secondary device.

Tristating the transmitter can also be performed automatically by the EUSART by using AUTOTRI in EUSARTn_CFG0. When AUTOTRI is set, the EUSART automatically tristates TX line whenever the transmitter is idle, and enables transmitter output when the transmitter goes active. If AUTOTRI is set TXTRI is always read as 0.

Note: Another way to tristate the transmitter is to enable wired-and or wired-or mode in GPIO. For wired-and mode, outputting a 1 will be the same as tristating the output, and for wired-or mode, outputting a 0 will be the same as tristating the output. This can only be done on buses with a pull-up or pull-down resistor respectively.

21.3.3.9.2 Single Data-link with External Driver

Some communication schemes, such as RS-485 rely on an external driver. Here, the driver has an extra input which enables it, and instead of tristating the transmitter when receiving data, the external driver must be disabled.

This can be done manually by assigning a GPIO to turn the driver on or off.

Figure 21.12. EUSART Half Duplex Communication with External Driver shows an example configuration using an external driver.

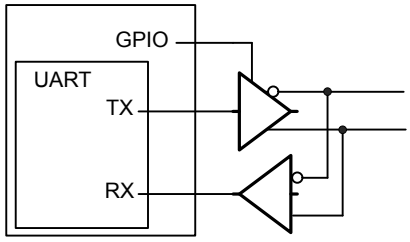


Figure 21.12. EUSART Half Duplex Communication with External Driver

21.3.3.9.3 Two Data-links

Some limited devices only support half duplex communication even though two data links are available. In this case software is responsible for making sure data is not transmitted when incoming data is expected.

21.3.3.10 Hardware Flow Control

Hardware flow control can be used to hold off the link partner's transmission until receive FIFO space is available. Use RTSPEN in GPIO_EUSARTn_ROUTEEN to enable RTS pin (CTS is an input so its pin is enabled by default). Port and Pin selection for RTS and CTS can be done in GPIO_EUSARTn_RTSROUTE/CTSROUTE. RTS is an out going signal which indicates that receive FIFO space is available to receive a frame. The link partner is being requested to send its data when RTS is active. RTS activation can be made dependent on how much space is available in the receive FIFO using RTSRXFW in EUSARTn_CFG1. For debug use set DBGHALT in EUSARTn_CFG1 which will force the RTS to request one frame from the link partner when the CPU core single steps. RTS is deactivated when RX is disabled.

CTS is an incoming signal to stop the next TX data from going out. CTS indicates that the link partner has receive FIFO space available, and the local transmitter is clear to send. When CTS deactivates in the middle of a frame, the frame currently being transmitted is completed before stopping. CTS operation needs to be enabled using CFG1.CTSEN.

The RTS and CTS are active low by default, but their polarity can be changed with RTSINV and CTSINV in EUSARTn_CFG1 respectively.

21.3.3.11 Debug Halt

During single stepping, debug halt feature allows halting EUSART frame reception by deactivating RTS when the core is halted and continuing frame reception by activating RTS when the core is unhalted. EUSART debug halt can be enabled by setting DBGHALT in EUSART_CFG1 to '1'. EUSART receiver must be enabled for debug halting.

When EUSART_CFG1.DBGHALT is not set or EUSART_CFG1.DBGHALT is set but chip halt is low, RTS is only dependent on the receive FIFO having space available to receive at least number of frames given by EUSART_CFG1.RTSRXFW settings.

When EUSART_CFG1.DBGHALT is set, RTS will remain deactivated as long as chip halt is high. When a low pulse is detected on chip halt while DBGHALT is set, RTS will be activated if receive FIFO has space available to receive at least number of frames given by EUSART_CFG1.RTSRXFW settings and no frame is being received. RTS will be deactivated again if chip halt goes back to high and receiver starts receiving a new frame or if receive FIFO does not have space available to receive at least number of frames given by EUSART_CFG1.RTSRXFW settings. This behavior allows single stepping to pulse the chip halt low for a cycle, and receive the next frame.

Please note that, if chip halt remains low for a short duration after RX is enabled while DBGHALT is set, initial low value of chip halt will be treated the same as a low pulse on chip halt.

As incoming frame is always received until receive FIFO is full regardless of the state of DBGHALT or chip halt, the link partner must stop transmitting when RTS is deactivated, or the receive FIFO could overflow.

All data in the transmit FIFO is sent out even when chip halt is asserted; therefore, DMA will need to be set to stop sending EUSART TX data during chip halt.

21.3.3.12 PRS-triggered Transmissions

If a transmission must be started on an event with very little delay, the PRS system can be used to trigger the transmission. The PRS channel to use as a trigger can be selected using PRS_CONSUMER_EUSARTn_TRIGGER.PRSSEL in the PRS block. When a positive edge is detected on PRS signal, the receiver is enabled if RXTEN in EUSARTn_TRIGCTRL is set, and the transmitter is enabled if TXTEN in EUSARTn_TRIGCTRL is set. Only one signal input is supported by the EUSART.

21.3.3.13 PRS RX Input

The EUSART can be configured to receive data directly from a PRS channel by setting RXPRSEN in EUSARTn_CFG1. The PRS channel used is selected using PRS_CONSUMER_EUSARTn_RX.PRSSEL in the PRS block.

21.3.3.14 DMA Support

The EUSART has full DMA support. The DMA controller can write to the transmit FIFO using the register EUSARTn_TXDATA, and it can read from the receive FIFO using the register EUSARTn_RXDATA. This enables 9 bit data + control/status bits transfers both to and from the EUSART.

A request for the DMA controller to read from the EUSART receive buffer can come from the following source:

- Receive FIFO is loaded with at least number of frames set by RXFIW.

A write request can come from the following source:

- Transmit FIFO has space for at least number of frames set by TXFIW

In some cases, it may be sensible to temporarily stop DMA read access to the EUSART when an error such as parity or framing error has occurred. This is enabled by setting ERRSDMA in EUSARTn_CFG0.

21.3.3.15 Interrupts

The interrupts generated by the EUSART are combined into two interrupt vectors. Interrupts related to reception are assigned to one interrupt vector, and interrupts related to transmission are assigned to the other. Separating the interrupts in this way allows different priorities to be set for transmission and reception interrupts.

The transmission interrupt vector groups the transmission-related interrupts generated by the following interrupt flags:

- TXC
- TXFL
- TXOF
- CCF
- TXIDLE

The reception interrupt on the other hand groups the reception-related interrupts, triggered by the following interrupt flags:

- RXFL
- RXFULL
- RXOF
- RXUF
- PERR
- FERR
- MPAF
- START
- SIGF
- AUTOBAUDDONE
- RXTO

If EUSARTn interrupts are enabled, an interrupt will be made if one or more of the interrupt flags in EUSARTn_IF and their corresponding bits in EUSARTn_IEN are set. All interrupts can serve as wake up interupts if enabled.

21.3.3.16 PRS

All PRS inputs are synchronized to the peripheral clock (clk_per) for the EUSART instance.

- RX_PRS: Input goes to RX module for data reception if RXPRSEN set to 1
- TRIGGER_PRS: Can be used to Enable TX and/or RX if TXTEN/RXTEN are set to 1

All PRS outputs come from clk_per flops.

- PRS_TX: Same as TX output of EUSART
- PRS_TXC: Same as the TXC status flag
- PRS_IRDA_TX: Registered (peripheral clock) version of IRHF output
- PRS_RXFL: Set if (EXFCNT > RXFIW)
- PRS_RTS: Same as RTS output of EUSART

21.3.3.17 IrDA Operation

The EUSART supports IrDA operation using both HF and LF clocks. For IrDA with HF clock, the controls are given in IRHFCFG register. For IrDA with LF clock (32.768 kHz), only RX is supported and the controls are given in IRLFCFG register. Note that OVS must be disabled for LF operation.

Note: Note that break generation/ detection feature is not supported while IrDA is enabled, i.e., either IRHFEN or IRLFEN is set.

21.3.3.17.1 IRHF

The IRHF modulator implements the physical layer of the IrDA specification, which is necessary for communication over IrDA. The modulator takes the signal output from the EUSART module, and modulates it before it leaves the EUSART. In the same way, the input signal is demodulated before it enters the actual EUSART module. The modulator implements the original Rev. 1.0 physical layer and one high speed extension which supports speeds from 2.4 kbps to 1.152 Mbps.

The data from and to the EUSART is represented in a NRZ (Non Return to Zero) format, where the signal value is at the same level through the entire bit period. For IrDA, the required format is RZI (Return to Zero Inverted), a format where a “1” is signalled by holding the line low, and a “0” is signalled by a short high pulse. An example is given in Figure 21.13. EUSART Example RZI Signal for a Given EUSART Frame.

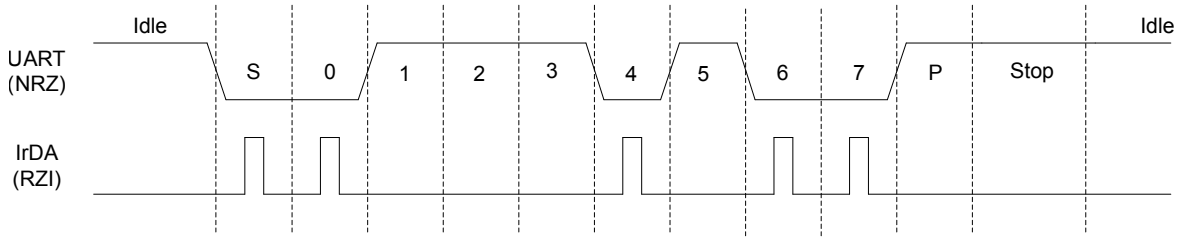


Figure 21.13. EUSART Example RZI Signal for a Given EUSART Frame

The IrDA HF module is enabled by setting IRHFEN in IRHFCFG. The EUSART transmitter output and receiver input is then routed through the IrDA HF modulator.

The width of the pulses generated by the IrDA HF modulator is set by configuring IRHFPW in IRHFCFG register. Four pulse widths are available, each defined relative to the configured bit period as listed in Table 21.10. EUSART IrDA Pulse Widths.

Table 21.10. EUSART IrDA Pulse Widths

IRHFPW	Pulse width OVS=0	Pulse width OVS=1	Pulse width OVS=2	Pulse width OVS=3
00	1/16	1/8	1/6	1/4
01	2/16	2/8	2/6	N/A
10	3/16	3/8	N/A	N/A
11	4/16	N/A	N/A	N/A

By default, no filter is enabled in the IrDA HF demodulator. A filter can be enabled by setting IRHFFILT in IRHFCFG. When the filter is enabled, an incoming pulse has to last for 5 consecutive clock cycles to be detected by the IrDA demodulator. When the filter is enabled, the minimum clock frequency required is based on the baud rate and OVS chosen. The frequency requirements are listed in table Table 21.11. EUSART IrDA IRHFFILT=1, Min Clock Frequency Requirement (MHz).

Table 21.11. EUSART IrDA IRHFFILT=1, Min Clock Frequency Requirement (MHz)

OVS	2.4 kb/s	9.6 kb/s	19.2 kb/s	38.4 kb/s	57.6 kb/s	115.2 kb/s	0.576 Mb/s	1.152 Mb/s
3 (x4)	1.0 MHz	1.0 MHz	1.0 MHz	1.0 MHz	1.4 MHz	2.8 MHz	13.8 MHz	27.6 MHz
2 (x6)	1.0 MHz	1.0 MHz	1.0 MHz	1.4 MHz	2.1 MHz	4.1 MHz	20.7 MHz	41.5 MHz
1 (x8)	1.0 MHz	1.0 MHz	1.0 MHz	1.0 MHz	1.4 MHz	2.8 MHz	13.8 MHz	27.6 MHz
0 (x16)	1.0 MHz	1.0 MHz	1.0 MHz	1.0 MHz	1.1 MHz	2.2 MHz	11.1 MHz	22.1 MHz

Note that by default, the idle value of the EUSART data signal is high. This means that the IrDA modulator generates negative pulses, and the IrDA demodulator expects negative pulses. To make the IrDA module use RZI signalling, both TXINV and RXINV in EUSARTn_CFG0 must be set.

Since the incoming signal is only sampled on positive clock edges, the width of the incoming pulses must be at least two clk_per periods wide for reliable detection by the receiver.

21.3.3.17.2 IRLF

IRLF only supports RX operation. This feature will stay operational even in all supported energy modes. It is possible to cause a wake up when a certain frame is received and then switch to IRHF if TX is required. IRLFEN in IRLFCFG must be set for this to work.

21.3.3.18 DALI Operation

EUSART supports DALI (Digitally Addressable Lighting Interface) operation using both HF and LF clocks down to energy mode EM1. DALICFG register holds the configuration for DALI operation. Once DALIEN is set to 1 in DALICFG, DALI mode gets enabled. DALI is available only in EUSART asynchronous mode, i.e., SYNC field in CFG0 register must be set to 0.

21.3.3.18.1 DALI Frame Format

DALI frames are Manchester encoded. Details:

- The TX/RX lines are high when idle.
- The Start bit is Manchester encoded to logic 1.
- Data bits are progammable (from 8-32) and are also Manchester encoded.
- Each DALI frame has two Stop bits, which are represented by a high on the line (same as idle status).
- DALI frame does not have parity bits.

DALI TX and RX data bits are independently programmable (each being from 8 to 32) using DALITXDATABITS and DALIRXDATABITS in the DALICFG register. The DALI bus is half-duplex so everything transmitted on the EUSART DALI TX pin will also show up on EUSART DALI RX pin. In order to ensure that whatever is sent out by EUSART TX is not sampled by EUSART RX, RX enable gets automatically masked to low during an active EUSART transmission. Once the transmission completes, the masking is removed (so if RX was enabled, removing the masking will result in RX getting enabled again and if RX was not enabled, RX enable will stay low even after removing the masking. This feature can be overridden for debugging by setting DALIRXENDT to 1 in DALICFG register. In that case, RX enable will not get masked during EUSART TX (i.e., if EUSART RX was already enabled, it will stay enabled during EUSART TX operation) and whatever is sent out will also be received by EUSART itself. In this case, it is important to set DALIRXDATABITS the same as DALITXDATABITS.

If LOOPBK in CFG0 is set to 1 in order to do loopback testing for DALI, then in this case DALIRXENDT should be set to 1 and DALIRXDATABITS should be set the same as DALITXDATABITS. Advantage of DALI loopback testing is that it can be done without connecting to an actual DALI bus since the data is looped back in from the TX pad.

21.3.3.18.2 DALI Baud Rate Formula

The EUSART clock defines DALI transmission and reception data rate. DALI mode supports any data rate from 300 baud up to 115.2 kbaud. DALI has a baudrate formula that differs from the standard EUSART UART mode baud rate formula. Whenever DALIEN in DALICFG is set to 1, then the DALI mode is enabled and the baud rate is given by Figure 21.14. DALI Baud Rate:

$$br = f_{EUSARTn}/(\text{oversample} \times 2 \times (1 + \text{EUSARTn_CLKDIV}/256))$$

Figure 21.14. DALI Baud Rate

where $f_{EUSARTn}$ is the peripheral clock frequency and oversample is the oversampling rate as defined by OVS in EUSARTn_CFG0.

Given a desired baud rate brdesired, the clock divider EUSARTn_CLKDIV can be calculated by using Figure 21.15. DALI Desired Baud Rate DIV:

$$\text{EUSARTn_CLKDIV} = 256 \times (f_{EUSARTn}/(\text{oversample} \times 2 \times \text{brdesired}) - 1)$$

Figure 21.15. DALI Desired Baud Rate DIV

The EUSART DALI mode can tolerate a 10% variation in baud rate as required in the DALI standard.

21.3.3.18.3 DALI Features

DALI mode is enabled by setting DALIEN to 1 in DALICFG register. A limited set of UART mode features are supported in DALI mode. Deatils below capture the feature set in DALI mode:

- A framing error interrupt (FERRIF) is generated if 2 stop bits are not received at the end of the frame. The received frame is still loaded into the RXDATA register.
- RXDATA and RXDATAP registers serve as RX FIFO. All 16 bits in these registers are treated as data in DALI mode. When DALI is disabled (DALIEN=0) while CFG0.SYNC=0, then these registers contain both data (max 9 bits) and also control information (PERR, FERR).
- When DALIEN is set to 1, then all 16 bits in TXDATA are treated as data. When DALIEN is 0 (DALI mode disabled) and CFG0.SYNC=0, then TXDATA contains both data bits and control bits.
- TXDATA and RXDATA registers are 16 bits wide but the DALI TX and RX data bits can be programmed to be anywhere between 8 and 32. In order to support data bits greater than 16 bits, special programming model applies. Please see the programming model section for details.
- SKIPERRF feature in CFG0 register is not supported in DALI mode.
- ERRSDMA, ERRSRX and ERRSTX features in CFG0 register are not supported in DALI mode.
- Multiprocessor mode is not supported in DALI mode.
- TX-tristating is not supported in DALI mode. The TX pin is driven high when idle.
- Collision detection is not supported in DALi mode.
- Autobaud detection is not supported in DALI mode.
- Hardware flow control and DBGHALT are not supported in DALI mode.
- RXTIMEOUT and TXDELAY features are not supported in DALI mode.
- Start frame feature is not supported in DALI mode.
- All settings in FRAMECFG are not used in DALI mode. DALICFG register settings are used instead.
- RX blocking feature is not supported in DALI mode.
- IrDA feature (IRLFCFG, IRHFCFG) cannot be enabled while DALI mode is enabled.

21.3.4 Synchronous Operation

Synchronous mode shares some common features with asynchronous mode such as: Loopback, inversion of RX/TX, MSBF, TX/RX Interrupt watermarks. They both share the EUSARTn_TRIGCTRL, CMD, TXDATA, RXDATA, STATUS and Interrupt registers.

In synchronous mode, EUSART can be configured to work either as a main (clock driver) or secondary (clock receiver) interface through EUSARTn_CFG2.MASTER.

21.3.4.1 Frame Format

The frames used in synchronous mode need no start and stop bits since a single clock is available to all parts participating in the communication. Parity bits cannot be used in synchronous mode.

The EUSART supports frame lengths of 8 to 16 bits per frame. Larger frames can be simulated by transmitting multiple smaller frames, i.e. a 22 bit frame can be sent using two 11-bit frames, and a 24 bit frame can be generated by transmitting three 8-bit frames. The number of bits in a frame is set using DATABITS in EUSARTn_FRAMECFG.

The frames in synchronous mode are by default transmitted with the least significant bit first like in asynchronous mode. The bit-order can be reversed by setting MSBF in EUSARTn_CFG0.

The frame format used by the transmitter can be inverted by setting TXINV in EUSARTn_CFG0, and the format expected by the receiver can be inverted by setting RXINV, also in EUSARTn_CFG0.

21.3.4.2 Clock Generation

The bit-rate in synchronous mode is given by Figure 21.16. EUSART Synchronous Mode Bit Rate. The clock division is derived from EUSARTn_CFG2.SDIV, which is applicable when acting as a Main interface only.

$$br = f_{EUSARTCLK}/(1 + EUSARTn_CFG2.SDIV)$$

where $f_{EUSARTCLK}$ is the peripheral clock coming from the CMU

Figure 21.16. EUSART Synchronous Mode Bit Rate

Given a desired baud rate brdesired, the clock divider EUSARTn_CFG2.SDIV can be calculated using Figure 21.17. EUSART Synchronous Mode Clock Division Factor

$$EUSARTn_CFG2.SDIV = (f_{EUSARTCLK}/brdesired - 1)$$

where $f_{EUSARTCLK}$ is the peripheral clock coming from the CMU

Figure 21.17. EUSART Synchronous Mode Clock Division Factor

On every clock edge data on the data lines, MOSI and MISO, is either set up or sampled. When CLKPHA in EUSARTn_CFG2 is cleared, data is sampled on the leading clock edge and set-up is done on the trailing edge. If CLKPHA is set however, data is set-up on the leading clock edge, and sampled on the trailing edge. In addition to this, the polarity of the clock signal can be changed by setting CLKPOL in EUSARTn_CFG2, which also defines the idle state of the clock. This results in four different modes which are summarized in Table 21.12. EUSART SPI Modes. Figure 21.18. EUSART SPI Timing shows the resulting timing of data set-up and sampling relative to the bus clock.

Table 21.12. EUSART SPI Modes

SPI mode	CLKPOL	CLKPHA	Leading edge	Trailing edge
0	0	0	Rising, sample	Falling, set-up
1	0	1	Rising, set-up	Falling, sample
2	1	0	Falling, sample	Rising, set-up
3	1	1	Falling, set-up	Rising, sample

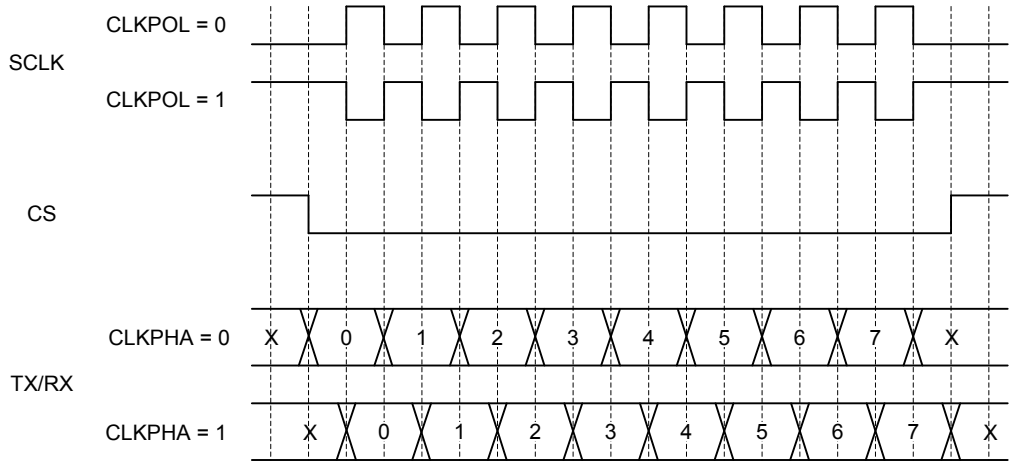


Figure 21.18. EUSART SPI Timing

The RX overflow interrupt flag, RXOF, is set at the end of the overflow frame if the receive FIFO is full. When a transfer has been performed, interrupt flags TXC are updated on the first setup clock edge of the succeeding frame, or when CS is deasserted.

21.3.4.3 Main SPI Interface Mode (Clock Driver)

EUSART operating as a main SPI interface is available only in EM0/EM1 with an HF clock source selected. When configured as a main interface, the EUSART is in full control of the data flow on the synchronous bus. When operating in full duplex mode, the secondary devices cannot transmit data to the main device without the main device transmitting to the secondary. The main device outputs the bus clock on SCLK.

Communication starts whenever there is data in the transmit FIFO and the transmitter is enabled. The EUSART clock then starts, and the main device shifts bits out from the transmit shift register using the internal clock.

When there are no more frames in the transmit FIFO and the transmit shift register is empty, the clock stops, and communication ends. When the receiver is enabled, it samples incoming data when the transmitter transmits data. Operation of the RX and TX FIFOs is as in asynchronous mode.

The EUSARTn_CMD.RXBLOCKEN command can be used to block incoming RX data from pushing into RX FIFO.

21.3.4.4 Operation of CS Pin

When operating as a main interface, the CS pin can have one of two functions, or it can be disabled by clearing CSPEN in GPIO_EUSARTn_ROUTEEN register.

If CS is disabled and there is a need to disable TX during the operation, then TX should be disabled at the end of transaction only indicated by the trigger of TXC interrupt to maintain synchronicity between TX and RX and avoid causing sudden stop to secondary devices.

If CS is configured as an output, it can be used to automatically generate a chip select for a single secondary device by setting AUTOCS in EUSARTn_CFG2. If AUTOCS is set, CS is activated before a transmission begins, and deactivated after the last bit has been transmitted and there is no more data in the transmit FIFO.

The time duration between assertion of CS and the start of transmission can be controlled using CSSETUP in EUSARTn_TIMINGCFG. If new data is ready for transmission before CS is deasserted, the data is sent without deasserting CS in between. CSHOLD in EUSARTn_TIMINGCFG keeps CS asserted after the end of frame for the number of baud-times specified.

By default, CS is active low, but its polarity can be inverted by setting CSINV in EUSARTn_CFG2.

21.3.4.5 AUTOTX

The main device on a synchronous bus is required to transmit data (send a clock) to a secondary device in order to receive data from that device. In some cases, only a few words are transmitted and a lot of data is then received from the secondary device. In that case, one solution is to keep feeding the TX with data to transmit, but that consumes system bandwidth. Instead AUTOTX can be used.

When AUTOTX in EUSARTn_CFG2 is set and TX FIFO is filled with initial data, EUSART will fully transmit the loaded data and then continue transmitting the last sent bit as long as there is available space in the RX FIFO for the chosen frame size. This happens even though there is no more data in the TX FIFO. The TX underflow interrupt flag TXUF in EUSARTn_IF is set on the first word that is transmitted which does not contain valid data.

During AUTOTX the EUSART will always send the previous sent bit, thus reducing the number of transitions on the TX output. So if the last bit sent was a 0, 0's will be sent during AUTOTX and if the last bit sent was a 1, 1's will be sent during AUTOTX.

21.3.4.6 Secondary SPI Interface Mode (Clock Receiver)

When the EUSART is in secondary interface mode, data transmission is not controlled by the EUSART, but by an external main SPI device. The EUSART is therefore not able to initiate a transmission, and has no control over the number of bytes written to the external main device.

The output and input to the EUSART are also swapped when in secondary mode, making the receiver take its input from TX (MOSI) and the transmitter drive RX (MISO).

To transmit data when in secondary mode, the device must load data into the transmit FIFO and enable the transmitter. The data will remain in the EUSART until the main device starts a transmission by pulling the CS input low and transmitting data. For every frame transmitted from main to secondary device, a frame is transferred from secondary to main as well.

If the transmitter is enabled in synchronous secondary mode and the main device starts transmission of a frame, the underflow interrupt flag TXUF in EUSARTn_IF will be set if no data is available for transmission. At the same time, the secondary device will transmit the default TX data, which can be set through EUSARTn_DTXDATCFG for the current and subsequent frames until the FIFO is filled. Note that when TX is enabled (with or without data in TXFIFO) in the middle of transaction, TXUF can be triggered if it's transmitting default TX data.

Similar to when operating as a synchronous main interface, The EUSARTn_CMD.RXBLOCKEN command can be used to block incoming RX data from pushing into RX FIFO.

If the secondary device needs to control its own chip select signal, this can be achieved by clearing CSPEN in the GPIO_EUSARTn_ROUTEEN register. The internal chip select signal can then be controlled through CSINV in the CTRL register. The chip select signal will be CSINV inverted, i.e. if CSINV is cleared, the chip select is active and vice versa. In such cases, SCLK could arrive anytime that the device doesn't have prior notification from CS. Hence, EUSARTn_CFG2.FORCELOAD bit can be used to control how the device transmits the first dataword. If this bit is not set, the next outgoing dataword will be a DEFAULT TX data even if the FIFO had been loaded before SCLK arrives, followed by the loaded TX data. EUSARTn_IF.LOADERR Interrupt will never be triggered when EUSARTn_CFG2.FORCELOAD is not set. If this bit is set, as soon as the transmitter becomes ready the shift register will be loaded immediately and send data once SCLK arrives. The transmitter becomes ready when TX is enabled and TXFIFO is filled. On top of that, at word-boundary it will automatically trigger setup window check against the programmed EUSARTn_TIMINGCFG.SETUPWINDOW, which specifies the minimum duration (in APB bus clock cycles) between transmitter becoming ready and the incoming SCLK. If the measured duration is less than SETUPWINDOW bus clock cycles, an EUSARTn_IF.LOADERR Interrupt will be triggered. Besides, if the transmitter is enabled or disabled or the empty TXFIFO is loaded not at word-boundary during a transaction (SCLK is toggling), LOADERR will also be triggered immediately without checking for SETUPWINDOW. It's recommended that the transmitter should be ready while SCLK is idling or at word-boundary and at sufficient margin before SCLK toggles. Once LOADERR Interrupt is set, it may require to reset the secondary interface by disabling the module and re-enabling it because the transmitted data could be un-determistic.

21.3.5 Debug Halt

When DBGHALT in EUSART_CFG1 is clear, RTS is only dependent on the RX FIFO having space available to receive data. Incoming data is always received until both the RX FIFO is full and the RX shift register is full regardless of the state of DBGHALT or chip halt. Additional incoming data is discarded. When DBGHALT is set, RTS deasserts on RX FIFO full or when chip halt is high. However, a low pulse detected on chip halt will keep RTS asserted when no frame is being received. At the start of frame reception, RTS will deassert if chip halt is high and DBGHALT is set. This behavior allows single stepping to pulse the chip halt low for a cycle, and receive the next frame. The link partner must stop transmitting when RTS is deasserted, or the RX FIFO could overflow. All data in the transmit FIFO is sent out even when chip halt is asserted; therefore, the DMA will need to be set to stop sending the EUSART TX data during chip halt.

21.3.6 PRS-triggered Transmissions

If a transmission must be started on an event with very little delay, the PRS system can be used to trigger the transmission. The PRS channel to use as a trigger can be selected using TSEL in EUSARTn_TRIGCTRL. When a positive edge is detected on this signal, the receiver is enabled if RXTEN in EUSARTn_TRIGCTRL is set, and the transmitter is enabled if TXTEN in EUSARTn_TRIGCTRL is set. Only one signal input is supported by the EUSART.

AUTOTXTEN can also be combined with TXTEN to make the EUSART transmit a command to the external device prior to clocking out data. To do this, disable TX using the TXDIS command, load the TX FIFO with the command and enable AUTOTXTEN and TXTEN. When the selected PRS input goes high, the EUSART will now transmit the loaded command, and then continue clocking out while both the PRS input is high and there is room in the RX FIFO

21.3.7 PRS RX Input

The EUSART can be configured to receive data directly from a PRS channel by setting RXPRSEN in EUSARTn_CFG1. The PRS channel used is selected in the PRS module. This way, for example, a differential RX signal can be input to the ACMP and the output routed via PRS to the EUSART.

21.3.8 PRS CLK Input

The EUSART can be configured to receive clock directly from a PRS channel by setting CLKPRSEN in EUSARTn_CFG2. The PRS channel used is selected in the PRS module. This is useful in secondary synchronous mode and can together with RX PRS input be used to input data from PRS.

21.3.9 DMA Support

The EUSART has full DMA support. The DMA controller can write to the transmit FIFO using the registers EUSARTn_TXDATA and it can read from the receive FIFO using the registers EUSARTn_RXDATA. This enables single byte transfers, 9 bit data + control/status bits, double byte and double byte + control/status transfers both to and from the EUSART.

A request for the DMA controller to read from the EUSART receive FIFO can come from the following source:

- Receive FIFO level satisfying EUSARTn_CFG1.RXFIW setting

A write request can come from one of the following sources:

- Transmit FIFO level satisfying EUSARTn_CFG1.TXFIW setting.

Even though there are two sources for write requests to the DMA, only one should be used at a time, since the requests from both sources are cleared even though only one of the requests are used.

In some cases, it may be sensible to temporarily stop DMA access to the EUSART when an error such as a framing error has occurred. This is enabled by setting ERRSDMA in EUSARTn_CFG0.

Note: For Synchronous mode full duplex operation, if both receive FIFO and transmit FIFO are served by DMA, to make sure receive FIFO is not overflowed the settings below should be followed.

- The DMA channel that serves receive FIFO should have higher priority than the DMA channel that serves transmit FIFO.
- IGNORESREQ should be set for both DMA channel.

21.4 EUSART Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	EUSART_IPVERSION	R	IP Version ID
0x004	EUSART_EN	RW ENABLE	Enable Register
0x008	EUSART_CFG0	RW CONFIG	Configuration 0 Register
0x00C	EUSART_CFG1	RW CONFIG	Configuration 1 Register
0x010	EUSART_CFG2	RW CONFIG	Configuration 2 Register
0x014	EUSART_FRAMECFG	RW CONFIG	Frame Format Register
0x018	EUSART_DTXDATCFG	RW CONFIG	Default TX DATA Register
0x01C	EUSART_IRHFCFG	RW CONFIG	HF IrDA Mod Config Register
0x020	EUSART_IRLFCFG	RW CONFIG	LF IrDA Pulse Config Register
0x024	EUSART_TIMINGCFG	RW CONFIG	Timing Register
0x028	EUSART_STARTFRAMECFG	RW CONFIG	Start Frame Register
0x02C	EUSART_SIGFRAMECFG	RW CONFIG	Signal Frame Register
0x030	EUSART_CLKDIV	RWH LFSYNC	Clock Divider Register
0x034	EUSART_TRIGCTRL	RW LFSYNC	Trigger Control Register
0x038	EUSART_CMD	W LFSYNC	Command Register
0x03C	EUSART_RXDATA	RH	RX Data Register
0x040	EUSART_RXDATAP	RH	RX Data Peek Register
0x044	EUSART_TXDATA	W	TX Data Register
0x048	EUSART_STATUS	RH	Status Register
0x04C	EUSART_IF	RWH INTFLAG	Interrupt Flag Register
0x050	EUSART_IEN	RW	Interrupt Enable Register
0x054	EUSART_SYNCBUSY	RH	Synchronization Busy Register
0x058	EUSART_DALICFG	RW CONFIG	DALI Config Register
0x1000	EUSART_IPVERSION_SET	R	IP Version ID
0x1004	EUSART_EN_SET	RW ENABLE	Enable Register
0x1008	EUSART_CFG0_SET	RW CONFIG	Configuration 0 Register
0x100C	EUSART_CFG1_SET	RW CONFIG	Configuration 1 Register
0x1010	EUSART_CFG2_SET	RW CONFIG	Configuration 2 Register
0x1014	EUSART_FRAMECFG_SET	RW CONFIG	Frame Format Register
0x1018	EUSART_DTXDATCFG_SET	RW CONFIG	Default TX DATA Register
0x101C	EUSART_IRHFCFG_SET	RW CONFIG	HF IrDA Mod Config Register
0x1020	EUSART_IRLFCFG_SET	RW CONFIG	LF IrDA Pulse Config Register
0x1024	EUSART_TIMINGCFG_SET	RW CONFIG	Timing Register
0x1028	EUSART_STARTFRA-MECFG_SET	RW CONFIG	Start Frame Register
0x102C	EUSART_SIGFRAMECFG_SET	RW CONFIG	Signal Frame Register
0x1030	EUSART_CLKDIV_SET	RWH LFSYNC	Clock Divider Register
0x1034	EUSART_TRIGCTRL_SET	RW LFSYNC	Trigger Control Register
0x1038	EUSART_CMD_SET	W LFSYNC	Command Register

Offset	Name	Type	Description
0x103C	EUSART_RXDATA_SET	RH	RX Data Register
0x1040	EUSART_RXDATAP_SET	RH	RX Data Peek Register
0x1044	EUSART_TXDATA_SET	W	TX Data Register
0x1048	EUSART_STATUS_SET	RH	Status Register
0x104C	EUSART_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1050	EUSART_IEN_SET	RW	Interrupt Enable Register
0x1054	EUSART_SYNCBUSY_SET	RH	Synchronization Busy Register
0x1058	EUSART_DALICFG_SET	RW CONFIG	DALI Config Register
0x2000	EUSART_IPVERSION_CLR	R	IP Version ID
0x2004	EUSART_EN_CLR	RW ENABLE	Enable Register
0x2008	EUSART_CFG0_CLR	RW CONFIG	Configuration 0 Register
0x200C	EUSART_CFG1_CLR	RW CONFIG	Configuration 1 Register
0x2010	EUSART_CFG2_CLR	RW CONFIG	Configuration 2 Register
0x2014	EUSART_FRAMECFG_CLR	RW CONFIG	Frame Format Register
0x2018	EUSART_DTXDATCFG_CLR	RW CONFIG	Default TX DATA Register
0x201C	EUSART_IRHFCFG_CLR	RW CONFIG	HF IrDA Mod Config Register
0x2020	EUSART_IRLFCFG_CLR	RW CONFIG	LF IrDA Pulse Config Register
0x2024	EUSART_TIMINGCFG_CLR	RW CONFIG	Timing Register
0x2028	EUSART_STARTFRA-MECFG_CLR	RW CONFIG	Start Frame Register
0x202C	EUSART_SIGFRAMECFG_CLR	RW CONFIG	Signal Frame Register
0x2030	EUSART_CLKDIV_CLR	RWH LFSYNC	Clock Divider Register
0x2034	EUSART_TRIGCTRL_CLR	RW LFSYNC	Trigger Control Register
0x2038	EUSART_CMD_CLR	W LFSYNC	Command Register
0x203C	EUSART_RXDATA_CLR	RH	RX Data Register
0x2040	EUSART_RXDATAP_CLR	RH	RX Data Peek Register
0x2044	EUSART_TXDATA_CLR	W	TX Data Register
0x2048	EUSART_STATUS_CLR	RH	Status Register
0x204C	EUSART_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2050	EUSART_IEN_CLR	RW	Interrupt Enable Register
0x2054	EUSART_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x2058	EUSART_DALICFG_CLR	RW CONFIG	DALI Config Register
0x3000	EUSART_IPVERSION_TGL	R	IP Version ID
0x3004	EUSART_EN_TGL	RW ENABLE	Enable Register
0x3008	EUSART_CFG0_TGL	RW CONFIG	Configuration 0 Register
0x300C	EUSART_CFG1_TGL	RW CONFIG	Configuration 1 Register
0x3010	EUSART_CFG2_TGL	RW CONFIG	Configuration 2 Register
0x3014	EUSART_FRAMECFG_TGL	RW CONFIG	Frame Format Register
0x3018	EUSART_DTXDATCFG_TGL	RW CONFIG	Default TX DATA Register
0x301C	EUSART_IRHFCFG_TGL	RW CONFIG	HF IrDA Mod Config Register
0x3020	EUSART_IRLFCFG_TGL	RW CONFIG	LF IrDA Pulse Config Register

Offset	Name	Type	Description
0x3024	EUSART_TIMINGCFG_TGL	RW CONFIG	Timing Register
0x3028	EUSART_STARTFRA-MECFG_TGL	RW CONFIG	Start Frame Register
0x302C	EUSART_SIGFRAMECFG_TGL	RW CONFIG	Signal Frame Register
0x3030	EUSART_CLKDIV_TGL	RWH LFSYNC	Clock Divider Register
0x3034	EUSART_TRIGCTRL_TGL	RW LFSYNC	Trigger Control Register
0x3038	EUSART_CMD_TGL	W LFSYNC	Command Register
0x303C	EUSART_RXDATA_TGL	RH	RX Data Register
0x3040	EUSART_RXDATAP_TGL	RH	RX Data Peek Register
0x3044	EUSART_TXDATA_TGL	W	TX Data Register
0x3048	EUSART_STATUS_TGL	RH	Status Register
0x304C	EUSART_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3050	EUSART_IEN_TGL	RW	Interrupt Enable Register
0x3054	EUSART_SYNCBUSY_TGL	RH	Synchronization Busy Register
0x3058	EUSART_DALICFG_TGL	RW CONFIG	DALI Config Register

21.5 EUSART Register Description

21.5.1 EUSART_IPVERSION - IP Version ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x4																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x4	R	**IP version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

21.5.2 EUSART_EN - Enable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	Disabling status when the module is disabled.			
0	EN	0x0	RW	**Module enable**
	Set to enable the module.			

21.5.3 EUSART_CFG0 - Configuration 0 Register

Offset	Bit Position																														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0						0x0	0x0	0x0		0x0			0x0			0x0	0x0			0x0			0x0			0x0	0x0	0x0	0x0	0x0
Access	RW	RW						RW	RW	RW		RW			RW			RW	RW			RW			RW			RW	RW	RW	RW	RW
Name	AUTOBAUDEN	MVDIS						ERRSTX	ERRSRX	ERRSDMA		SKIPPERRF			AUTOTRI			TXINV	RXINV			MSBF			OVS			MPAB	MPM	CCEN	LOOPBK	SYNC

Bit	Name	Reset	Access	Description
31	AUTOBAUDEN	0x0	RW	**AUTOBAUD detection enable**
	Detects the baud rate based on receiving a 0x55 frame (0x00 for IrDA). Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
30	MVDIS	0x0	RW	**Majority Vote Disable**
	Disable majority vote for 16x, 8x and 6x oversampling modes. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
29:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	ERRSTX	0x0	RW	**Disable TX On Error**
	When set, the transmitter is disabled on framing and parity errors in the receiver. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

Value	Mode	Description
0	DISABLE	Received framing and parity errors have no effect on transmitter
1	ENABLE	Received framing and parity errors disable the transmitter

Bit	Name	Reset	Access	Description
23	ERRSRX	0x0	RW	**Disable RX On Error**
	When set, the receiver is disabled on framing and parity errors. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

Value	Mode	Description
0	DISABLE	Framing and parity errors have no effect on receiver
1	ENABLE	Framing and parity errors disable the receiver

Bit	Name	Reset	Access	Description
22	ERRSDMA	0x0	RW	**Halt DMA Read On Error**
	When set, DMA read requests will be cleared on framing and parity errors. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

Value	Mode	Description
0	DISABLE	Framing and parity errors have no effect on DMA requests from the EUSART
1	ENABLE	DMA requests from the EUSART are blocked while the PERR or FERR interrupt flags are set

Bit	Name	Reset	Access	Description
21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	SKIPPERRF	0x0	RW	**Skip Parity Error Frames**
	When set, the receiver discards frames with parity errors. The PERR interrupt flag is still set. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

Bit	Name	Reset	Access	Description
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	AUTOTRI	0x0	RW	**Automatic TX Tristate**

When enabled, TXTRI is set by hardware whenever the transmitter is idle, and TXTRI is cleared by hardware when transmission starts. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Value	Mode	Description
0	DISABLE	The output on UARTn_TX when the transmitter is idle is defined by TXINV
1	ENABLE	UARTn_TX is tristated whenever the transmitter is idle

16:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14	TXINV	0x0	RW	**Transmitter output Invert**

The output from the EUSART transmitter can optionally be inverted by setting this bit.

Value	Mode	Description
0	DISABLE	Output from the transmitter is passed unchanged to UARTn_TX
1	ENABLE	Output from the transmitter is inverted before it is passed to UARTn_TX

13	RXINV	0x0	RW	**Receiver Input Invert**

Setting this bit will invert the input to the EUSART receiver.

Value	Mode	Description
0	DISABLE	Input is passed directly to the receiver
1	ENABLE	Input is inverted before it is passed to the receiver

12:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	MSBF	0x0	RW	**Most Significant Bit First**

Decides whether data is sent with the least significant bit first, or the most significant bit first.

Value	Mode	Description
0	DISABLE	Data is sent with the least significant bit first
1	ENABLE	Data is sent with the most significant bit first

9:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:5	OVS	0x0	RW	**Oversampling**

Sets the number of clock periods in a EUSART bit-period. More clock cycles gives better robustness, while less clock cycles gives better performance. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Value	Mode	Description
0	X16	16X oversampling
1	X8	8X oversampling
2	X6	6X oversampling
3	X4	4X oversampling
4	DISABLE	Disable oversampling (for LF operation)

4	MPAB	0x0	RW	**Multi-Processor Address-Bit**

Bit	Name	Reset	Access	Description
	Defines the value of the multi-processor address bit. An incoming frame with its 9th bit equal to the value of this bit marks the frame as a multi-processor address frame. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
3	MPM	0x0	RW	**Multi-Processor Mode**
	Multi-processor mode uses the 9th bit of the EUSART frames to tell whether the frame is an address frame or a data frame. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
	Value	Mode		Description
	0	DISABLE		The 9th bit of incoming frames has no special function
	1	ENABLE		An incoming frame with the 9th bit equal to MPAB will be loaded into the RX FIFO regardless of RXBLOCK and will result in the MPAB interrupt flag being set
2	CCEN	0x0	RW	**Collision Check Enable**
	Enables collision checking on data when operating in half duplex modus. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
	Value	Mode		Description
	0	DISABLE		Collision check is disabled
	1	ENABLE		Collision check is enabled. The receiver must be enabled for the check to be performed
1	LOOPBK	0x0	RW	**Loopback Enable**
	Allows the receiver to be connected directly to the EUSART transmitter for loopback and half duplex communication.			
	Value	Mode		Description
	0	DISABLE		The receiver is connected to and receives data from UARTn_RX
	1	ENABLE		The receiver is connected to and receives data from UARTn_TX
0	SYNC	0x0	RW	**Synchronous Mode**
	Determines whether the EUSART is operating in asynchronous or synchronous mode. When switching between SYNC and ASYNC mode, module Disablement is required.			
	Value	Mode		Description
	0	ASYNC		The USART operates in asynchronous mode
	1	SYNC		The USART operates in synchronous mode

21.5.4 EUSART_CFG1 - Configuration 1 Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0					0x0						0x0					0x0				0x0					0x0			0x0	0x0	0x0	0x0
Access	RW					RW						RW					RW				RW					RW			RW	RW	RW	RW
Name	RXFIW					RTSRXFW						TXFIW					RXPRSEN				SFUBRX					RXTIMEOUT			RTSINV	CTSEN	CTSINV	DBGHALT

Bit	Name	Reset	Access	Description
31:27	RXFIW	0x0	RW	**RX FIFO Interrupt Watermark**

Determines the interrupt and status level of the Receive FIFO. Also impacts RX DMA request.

Value	Mode	Description
0	ONEFRAME	RXFL status flag and IF are set when the RX FIFO has at least one frame in it.
1	TWOFRAMES	RXFL status flag and IF are set when the RX FIFO has at least two frames in it.
2	THREEFRAMES	RXFL status flag and IF are set when the RX FIFO has at least three frames in it.
3	FOURFRAMES	RXFL status flag and IF are set when the RX FIFO has at least four frames in it.
4	FIVEFRAMES	RXFL status flag and IF are set when the RX FIFO has at least five frames in it.
5	SIXFRAMES	RXFL status flag and IF are set when the RX FIFO has at least six frames in it.
6	SEVENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least seven frames in it.
7	EIGHTFRAMES	RXFL status flag and IF are set when the RX FIFO has at least eight frames in it.
8	NINEFRAMES	RXFL status flag and IF are set when the RX FIFO has at least nine frames in it.
9	TENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least ten frames in it.
10	ELEVENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least eleven frames in it.
11	TWELVEFRAMES	RXFL status flag and IF are set when the RX FIFO has at least twelve frames in it.
12	THIRTEENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least thriteen frames in it.
13	FOURTEENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least fourteen frames in it.
14	FIFTEENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least fifteen frames in it.
15	SIXTEENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least sixteen frames in it.
16	SEVENTEENFRAMES	RXFL status flag and IF are set when the RX FIFO has at least seventeen frames in it.

Bit	Name	Reset	Access	Description
	17	EIGHTEENFRAMES		RXFL status flag and IF are set when the RX FIFO has at least eighteen frames in it.
	18	NINETEENFRAMES		RXFL status flag and IF are set when the RX FIFO has at least nineteen frames in it.
	19	TWENTYFRAMES		RXFL status flag and IF are set when the RX FIFO has at least twenty frames in it.
	20	TWENTYONEFRAMES		RXFL status flag and IF are set when the RX FIFO has at least twentyone frames in it.
	21	TWENTYTWOFRAMES		RXFL status flag and IF are set when the RX FIFO has at least twentytwo frames in it.
	22	TWENTYTHREEFRAMES		RXFL status flag and IF are set when the RX FIFO has at least twentythree frames in it.
	23	TWENTYFOURFRAMES		RXFL status flag and IF are set when the RX FIFO has at least twentyfourframes in it.
	24	TWENTYFIVEFRAMES		RXFL status flag and IF are set when the RX FIFO has at least twentyfive frames in it.
	25	TWENTYSIXFRAMES		RXFL status flag and IF are set when the RX FIFO has at least twentysix frames in it.
	26	TWENTYSEVENFRAMES		RXFL status flag and IF are set when the RX FIFO has at least twentyseven frames in it.
	27	TWENTYEIGHTFRAMES		RXFL status flag and IF are set when the RX FIFO has at least twentyeight frames in it.
	28	TWENTYNINEFRAMES		RXFL status flag and IF are set when the RX FIFO has at least twentynine frames in it.
	29	THIRTYFRAMES		RXFL status flag and IF are set when the RX FIFO has at least thirty frames in it.
	30	THIRTYONEFRAMES		RXFL status flag and IF are set when the RX FIFO has at least thirtyone frames in it.
	31	THIRTYTWOFRAMES		RXFL status flag and IF are set when the RX FIFO has at least thirtytwo frames in it.
26:22	RTSRXFW	0x0	RW	**Request-to-send RX FIFO Watermark**

Set Request-to-send watermark level. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Value	Mode	Description
0	ONEFRAME	RTS is set if there is space for at least one more frame in the RX FIFO.
1	TWOFRAMES	RTS is set if there is space for at least two more frames in the RX FIFO.
2	THREEFRAMES	RTS is set if there is space for at least three more frames in the RX FIFO.
3	FOURFRAMES	RTS is set if there is space for four more frames in the RX FIFO.
4	FIVEFRAMES	RTS is set if there is space for five more frames in the RX FIFO.
5	SIXFRAMES	RTS is set if there is space for six more frames in the RX FIFO.
6	SEVENFRAMES	RTS is set if there is space for seven more frames in the RX FIFO.
7	EIGHTFRAMES	RTS is set if there is space for eight more frames in the RX FIFO.
8	NINEFRAMES	RTS is set if there is space for nine more frames in the RX FIFO.

Bit	Name	Reset	Access	Description
	9	TENFRAMES		RTS is set if there is space for ten more frames in the RX FIFO.
	10	ELEVENFRAMES		RTS is set if there is space for eleven more frames in the RX FIFO.
	11	TWELVEFRAMES		RTS is set if there is space for twelve more frames in the RX FIFO.
	12	THIRTEENFRAMES		RTS is set if there is space for thirteen more frames in the RX FIFO.
	13	FOURTEENFRAMES		RTS is set if there is space for fourteen more frames in the RX FIFO.
	14	FIFTEENFRAMES		RTS is set if there is space for fifteen more frames in the RX FIFO.
	15	SIXTEENFRAMES		RTS is set if there is space for sixteen more frames in the RX FIFO.
	16	SEVENTEENFRAMES		RTS is set if there is space for seventeen more frames in the RX FIFO.
	17	EIGHTEENFRAMES		RTS is set if there is space for eighteen more frames in the RX FIFO.
	18	NINETEENFRAMES		RTS is set if there is space for nineteen more frames in the RX FIFO.
	19	TWENTYFRAMES		RTS is set if there is space for twenty more frames in the RX FIFO.
	20	TWENTYONEFRAMES		RTS is set if there is space for twentyone more frames in the RX FIFO.
	21	TWENTYTWOFRAMES		RTS is set if there is space for twentytwo more frames in the RX FIFO.
	22	TWENTYTHREEFRAMES		RTS is set if there is space for twentythree more frames in the RX FIFO.
	23	TWENTYFOURFRAMES		RTS is set if there is space for twentyfour more frames in the RX FIFO.
	24	TWENTYFIVEFRAMES		RTS is set if there is space for twentyfive more frames in the RX FIFO.
	25	TWENTYSIXFRAMES		RTS is set if there is space for twentysix more frames in the RX FIFO.
	26	TWENTYSEVENFRAMES		RTS is set if there is space for twentyseven more frames in the RX FIFO.
	27	TWENTYEIGHTFRAMES		RTS is set if there is space for twentyeight more frames in the RX FIFO.
	28	TWENTYNINEFRAMES		RTS is set if there is space for twetynine more frames in the RX FIFO.
	29	THIRTYFRAMES		RTS is set if there is space for thirty more frames in the RX FIFO.
	30	THIRTYONEFRAMES		RTS is set if there is space for thirtyone more frames in the RX FIFO.
	31	THIRTYTWOFRAMES		RTS is set if there is space for thirtytwo more frames in the RX FIFO.
21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20:16	TXFIW	0x0	RW	**TX FIFO Interrupt Watermark**
	Determines the interrupt and status level of the transmit FIFO. Also impacts TX DMA request.			

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	ONEFRAME		TXFL status flag and IF are set when the TX FIFO has space for at least one more frame.
	1	TWOFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least two more frames.
	2	THREEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least three more frames.
	3	FOURFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least four more frames.
	4	FIVEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least five more frames.
	5	SIXFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least six more frames.
	6	SEVENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least seven more frames.
	7	EIGHTFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least eight more frames.
	8	NINEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least nine more frames.
	9	TENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least ten more frames.
	10	ELEVENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least eleven more frames.
	11	TWELVEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least twelve more frames.
	12	THIRTEENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least thriteen more frames.
	13	FOURTEENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least fourteen more frames.
	14	FIFTEENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least fifteen more frames.
	15	SIXTEENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least sixteen more frames.
	16	SEVENTEENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least seventeen more frames.
	17	EIGHTEENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least eighteen more frames.
	18	NINETEENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least nineteen more frames.
	19	TWENTYFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least twenty more frames.
	20	TWENTYONEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least twentyone more frames.
	21	TWENTYTWOFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least twentytwo more frames.
	22	TWENTYTHREEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least twentythree more frames.
	23	TWENTYFOURFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least twentyfour more frames.

Bit	Name	Reset	Access	Description
	24	TWENTYFIVEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least twentyfive more frames.
	25	TWENTYSIXFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least twentysix more frames.
	26	TWENTYSEVENFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least twentyseven more frames.
	27	TWENTYEIGHTFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least twentyeight more frames.
	28	TWENTYNINEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least twentynine more frames.
	29	THIRTYFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least thirty more frames.
	30	THIRTYONEFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least thirtyone more frames.
	31	THIRTYTWOFRAMES		TXFL status flag and IF are set when the TX FIFO has space for at least thirtytwo more frames.
15	RXPRSEN	0x0	RW	**PRS RX Enable**
	When set, the PRS channel selected as input to RX.			
14:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	SFUBRX	0x0	RW	**Start Frame Unblock Receiver**
	Set to unblock RX on Start frame reception. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
10:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	RXTIMEOUT	0x0	RW	**RX Timeout**
	When enabled, determines how long, in units of frame, RX needs to remain idle after a frame reception before RXTOIF gets set. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

Value	Mode	Description
0	DISABLED	
1	ONEFRAME	
2	TWOFRAMES	
3	THREEFRAMES	
4	FOURFRAMES	
5	FIVEFRAMES	
6	SIXFRAMES	
7	SEVENFRAMES	

3	RTSINV	0x0	RW	**Request-to-send Invert Enable**
	When set, the RTS pin polarity is inverted. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

Value	Mode	Description
0	DISABLE	The RTS pin is active low
1	ENABLE	The RTS pin is active high

2	CTSEN	0x0	RW	**Clear-to-send Enable**

Bit	Name	Reset	Access	Description
	When set, frames in the TX FIFO will not be sent until link partner asserts CTS. Any data in the TX shift register will continue transmitting, the next TX FIFO data will not load into the TX shift register. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
	Value	Mode		Description
	0	DISABLE		Ignore CTS
	1	ENABLE		Stop transmitting when CTS is inactive
1	CTSINV	0x0	RW	**Clear-to-send Invert Enable**
	When set, the CTS pin polarity is inverted. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
	Value	Mode		Description
	0	DISABLE		The CTS pin is active low
	1	ENABLE		The CTS pin is active high
0	DBGHALT	0x0	RW	**Debug halt**
	Set to halt operation when core is halted. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
	Value	Mode		Description
	0	DISABLE		Continue normal EUSART operation even if core is halted
	1	ENABLE		If core is halted, receive one frame and then halt reception by deactivating RTS. Next frame reception happens when the core is unhalted during single stepping.

21.5.5 EUSART_CFG2 - Configuration 2 Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																								0x0	0x0	0x1	0x0	0x0	0x0	0x0	0x0
Access	RW																								RW	RW	RW	RW	RW	RW	RW	RW
Name	SDIV																								FORCELOAD	CLKPRSEN	AUTOCS	AUTOTX	CSINV	CLKPHA	CLKPOL	MASTER

Bit	Name	Reset	Access	Description
31:24	SDIV	0x0	RW	**Sync Clock Div**
	Sets the clock rate for synchronous master operation only (To set the clock rate for asynchronous operation, see the CLKDIV field). Clock division value = SDIV + 1. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.			
23:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	FORCELOAD	0x0	RW	**Force Load to Shift Register**
	When this bit is set, if TXEN is already enabled, any loading of an empty FIFO will immediately load the data into the shift register. This bit is recommended to be used in 3-wire setting where CS is not used, or in a custom 4-wire mode where SCLK is in long idle state between two transactions with CS always in active mode (this typically happens between 2 transactions where the master holds SCLK idling to give additional time for the slave to load data). If FORCELOAD bit is not set, the next out-going frame will be a DEFAULT TX data, followed by the loaded TX data in TX FIFO. When FORCELOAD is set, it will automatically trigger setup window check against the programmed EUSARTn_TIMINGCFG.SETUPWINDOW, which specifies the minimum duration between empty fifo loading event and first encountered edge of SCLK. If the measured duration is less than SETUPWINDOW bus clock cycles, a EUSARTn_IF.LOADERRIF Interrupt will be triggered. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'SLAVE'.			
6	CLKPRSEN	0x0	RW	**PRS CLK Enable**
	When set, the PRS channel selected as input to CLK. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'SLAVE'.			
5	AUTOCS	0x1	RW	**Automatic Chip Select**
	When enabled, the output on CS will be activated one baud-period before transmission starts, and deactivated when transmission ends. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.			
4	AUTOTX	0x0	RW	**Always Transmit When RXFIFO Not Full**
	Transmits as long as RXFIFO is not full. If TX is empty, underflows are generated. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.			
3	CSINV	0x0	RW	**Chip Select Invert**
	Default value is active low. This affects both the selection of external slaves, as well as the selection of the microcontroller as a slave. Only applicable when CFG0.SYNC bit is set to 'SYNC'.			

Value	Mode	Description
0	AL	Chip select is active low
1	AH	Chip select is active high

Bit	Name	Reset	Access	Description
2	CLKPHA	0x0	RW	**Clock Edge for Setup/Sample**
	Determines where data is set-up and sampled according to the bus clock when in synchronous mode. Only applicable when CFG0.SYNC bit is set to 'SYNC'.			

Value	Mode	Description

Bit	Name	Reset	Access	Description
	0	SAMPLELEADING		Data is sampled on the leading edge and set-up on the trailing edge of the bus clock in synchronous mode
	1	SAMPLETRAILING		Data is set-up on the leading edge and sampled on the trailing edge of the bus clock in synchronous mode
1	CLKPOL	0x0	RW	**Clock Polarity**

Determines the clock polarity of the bus clock used in synchronous mode. Only applicable when CFG0.SYNC bit is set to 'SYNC'.

Value	Mode	Description
0	IDLELOW	The bus clock used in synchronous mode has a low base value
1	IDLEHIGH	The bus clock used in synchronous mode has a high base value

0	MASTER	0x0	RW	**Master mode**

Setting this bit to put EUSART to master mode. When unset, EUSART operates in slave mode. When changing between Master and Slave mode, module Disablement is required. Only applicable when CFG0.SYNC bit is set to 'SYNC'. When CFG0.SYNC bit is set to 'ASYNC', this bit must be set to 0.

Value	Mode	Description
0	SLAVE	Slave mode
1	MASTER	Master mode

21.5.6 EUSART_FRAMECFG - Frame Format Register

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																			0x1				0x0						0x2			
Access																			RW				RW						RW			
Name																			STOPBITS				PARITY						DATABITS			

Bit	Name	Reset	Access	Description
31:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	STOPBITS	0x1	RW	**Stop-Bit Mode**

Determines the number of stop-bits used. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Value	Mode	Description
0	HALF	The transmitter generates a half stop bit. Stop-bits are not verified by receiver
1	ONE	One stop bit is generated and verified
2	ONEANDAHALF	The transmitter generates one and a half stop bit. The receiver verifies the first stop bit
3	TWO	The transmitter generates two stop bits. The receiver checks the first stop-bit only

11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	PARITY	0x0	RW	**Parity-Bit Mode**

Determines whether parity bits are enabled, and whether even or odd parity should be used. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.

Value	Mode	Description
0	NONE	Parity bits are not used
2	EVEN	Even parity are used. Parity bits are automatically generated and checked by hardware.
3	ODD	Odd parity is used. Parity bits are automatically generated and checked by hardware.

7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	DATABITS	0x2	RW	**Data-Bit Mode**

Sets the number of data bits in a EUSART frame.

Value	Mode	Description
1	SEVEN	Each frame contains 7 data bits
2	EIGHT	Each frame contains 8 data bits
3	NINE	Each frame contains 9 data bits
4	TEN	Each frame contains 10 data bits
5	ELEVEN	Each frame contains 11 data bits

Bit	Name	Reset	Access	Description
	6	TWELVE		Each frame contains 12 data bits
	7	THIRTEEN		Each frame contains 13 data bits
	8	FOURTEEN		Each frame contains 14 data bits
	9	FIFTEEN		Each frame contains 15 data bits
	10	SIXTEEN		Each frame contains 16 data bits

21.5.7 EUSART_DTXDATCFG - Default TX DATA Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	RW															
Name																	DTXDAT															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	DTXDAT	0x0	RW	**Default TX DATA**
	This is the default transmitted data when the TXFIFO is empty. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'SLAVE'.			

21.5.8 EUSART_IRHFCFG - HF IrDA Mod Config Register

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0		0x0
Access																													RW	RW		RW
Name																													IRHFFILT	IRHFPW		IRHFEN

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	IRHFFILT	0x0	RW	**IrDA RX Filter**
	Set to enable filter on demodulator.			
	Value	Mode		Description
	0	DISABLE		No filter enabled
	1	ENABLE		Filter enabled. IrDA pulse must be high for at least 5 consecutive clock cycles to be detected
2:1	IRHFPW	0x0	RW	**IrDA TX Pulse Width**
	Configure the pulse width generated by the modulator as a fraction of the configured EUSART bit period.			
	Value	Mode		Description
	0	ONE		IrDA pulse width is 1/16 for OVS=0 and 1/8 for OVS=1
	1	TWO		IrDA pulse width is 2/16 for OVS=0 and 2/8 for OVS=1
	2	THREE		IrDA pulse width is 3/16 for OVS=0 and 3/8 for OVS=1
	3	FOUR		IrDA pulse width is 4/16 for OVS=0 and 4/8 for OVS=1
0	IRHFEN	0x0	RW	**Enable IrDA Module**
	Enable IrDA module and route EUSART signals through it. Used when EUSART has HF clock. Only applicable when CFG0.SYNC bit is set to 'ASYNC'			

21.5.9 EUSART_IRLFCFG - LF IrDA Pulse Config Register

Offset	Bit Position																															
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																IRLFEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	IRLFEN	0x0	RW	**Pulse Generator/Extender Enable**
	Filter EUSART output through pulse generator and the EUSART input through the pulse extender. Used for LF operation. Only applicable when CFG0.SYNC bit is set to 'ASYNC'			

21.5.10 EUSART_TIMINGCFG - Timing Register

Offset	Bit Position																															
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x5					0x0				0x0				0x0					0x0	
Access													RW					RW				RW				RW					RW	
Name													SETUPWINDOW					ICS				CSHOLD				CSSETUP					TXDELAY	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	SETUPWINDOW	0x5	RW	**Setup Window**

When CFG2.FOCELOAD is set, this defines the number of bus clock cycles that empty FIFO load or enabling of TX or disabling of TX must be performed before the sampling edge of SCLK at word-boundary to avoid load error. Word boundary is defined as followings:before the transaction starts, or between 2 transactions or the first bit between 2 datawords. If baud-rate is more than 5MHz, a value of 4 is recommended, any values smaller than that can be tried out but avoid using 0. If baud-rate is less than 5MHz, value of 5 is recommended, any values higher than 5 can be used but it may make the load error easy to occur. The recommended values for frequency bands should be sufficient to work all the time. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'SLAVE' and CFG2.FORCELOAD is set.

15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:12	ICS	0x0	RW	**Inter-Character Spacing**

Inter-character spacing after each TX frame while the TX FIFO is not empty. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.

Value	Mode	Description
0	ZERO	There is no space between charcters
1	ONE	Create a space of 1 baud-times between frames
2	TWO	Create a space of 2 baud-times between frames
3	THREE	Create a space of 3 baud-times between frames
4	FOUR	Create a space of 4 baud-times between frames
5	FIVE	Create a space of 5 baud-times between frames
6	SIX	Create a space of 6 baud-times between frames
7	SEVEN	Create a space of 7 baud-times between frames

11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:8	CSHOLD	0x0	RW	**Chip Select Hold**

Chip Select will be de-asserted after the end of frame transmission. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.

Value	Mode	Description
0	ZERO	CS is de-asserted half or 1 baud-time after the end of transmission depending on CLKPHASE equal to 1 or 0 respectively

Bit	Name	Reset	Access	Description
	1	ONE		CS is de-asserted 1 additional baud-time after the end of transmission
	2	TWO		CS is de-asserted 2 additional baud-times after the end of transmission
	3	THREE		CS is de-asserted 3 additional baud-times after the end of transmission
	4	FOUR		CS is de-asserted 4 additional baud-times after the end of transmission
	5	FIVE		CS is de-asserted 5 additional baud-times after the end of transmission
	6	SIX		CS is de-asserted 6 additional baud-times after the end of transmission
	7	SEVEN		CS is de-asserted 7 additional baud-times after the end of transmission
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	CSSETUP	0x0	RW	**Chip Select Setup**
	Chip Select will be asserted before the start of frame transmission. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.			
	Value	Mode		Description
	0	ZERO		CS is asserted half or 1 baud-time before the start of transmission depending on CLKPHASE equal to 1 or 0 respectively
	1	ONE		CS is asserted 1 additional baud-time before start of transmission
	2	TWO		CS is asserted 2 additional baud-times before start of transmission
	3	THREE		CS is asserted 3 additional baud-times before start of transmission
	4	FOUR		CS is asserted 4 additional baud-times before start of transmission
	5	FIVE		CS is asserted 5 additional baud-times before start of transmission
	6	SIX		CS is asserted 6 additional baud-times before start of transmission
	7	SEVEN		CS is asserted 7 additional baud-times before start of transmission
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	TXDELAY	0x0	RW	**TX Delay Transmission**
	Configurable delay before new rtansfers. Frames sent back-to-back are not delayed. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
	Value	Mode		Description
	0	NONE		Frames are transmitted immediately.
	1	SINGLE		Transmission of new frames is delayed by a single bit period.
	2	DOUBLE		Transmission of new frames is delayed by a two bit periods.
	3	TRIPPLE		Transmission of new frames is delayed by a three bit periods.

21.5.11 EUSART_STARTFRAMECFG - Start Frame Register

Offset	Bit Position																															
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0								
Access																								RW								
Name																								STARTFRAME								

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	STARTFRAME	0x0	RW	**Start Frame**
	When a frame matching STARTFRAME is received, the receiver detects that and STARTF interrupt flag is set. If SFUBRX is set, RXBLOCK is cleared and the start frame is loaded in to the RX FIFO. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

21.5.12 EUSART_SIGFRAMECFG - Signal Frame Register

Offset	Bit Position																															
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	SIGFRAME																															

Bit	Name	Reset	Access	Description
31:0	SIGFRAME	0x0	RW	**Signal Frame Value**
	When a frame matching SIGFRAME is detected by the receiver, SIGF interrupt flag is set. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

21.5.13 EUSART_CLKDIV - Clock Divider Register

Offset	Bit Position																															
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset										0x0																						
Access										RW																						
Name										DIV																						

Bit	Name	Reset	Access	Description
31:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:3	DIV	0x0	RW	**Fractional Clock Divider** Specifies the fractional clock divider. Setting AUTOBAUDEN in CFG1 register will overwrite the DIV field. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.
2:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

21.5.14 EUSART_TRIGCTRL - Trigger Control Register

Offset	Bit Position																															
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														AUTOTXTEN	TXTEN	RXTEN

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	AUTOTXTEN	0x0	RW	**AUTOTX Trigger Enable** When set, AUTOTX is enabled as long as the selected PRS channel has a high value. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'MASTER'.
1	TXTEN	0x0	RW	**Transmit Trigger Enable** When set, the positive edge of the selected PRS channel sets TXEN, enabling the transmitter.
0	RXTEN	0x0	RW	**Receive Trigger Enable** When set, the positive edge of the selected PRS channel sets RXEN, enabling the receiver.

21.5.15 EUSART_CMD - Command Register

Offset	Bit Position																													
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																							W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																							CLEARRX	CLEARTX	TXTRIDIS	TXTRIEN	RXBLOCKDIS	RXBLOCKEN	TXDIS	TXEN	RXDIS	RXEN

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	CLEARRX	0x0	W(nB)	**Clear RX FIFO**
	Set to clear RX FIFO. Note that the receiver must be disabled and poll for its status before issuing this command.			
8	CLEARTX	0x0	W(nB)	**Clear TX FIFO**
	Set to clear TX FIFO. Only applicable when CFG0.SYNC bit is set to 'ASYNC'. Note that before issuing this command, firmware should first set the transmitter disable bit (CMD.TXDIS) and then poll the transmitter enable status bit until cleared (STATUS.TXENS).			
7	TXTRIDIS	0x0	W(nB)	**Transmitter Tristate Disable**
	Disables tristating of the transmitter output.			
6	TXTRIEN	0x0	W(nB)	**Transmitter Tristate Enable**
	Tristates the transmitter output. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
5	RXBLOCKDIS	0x0	W(nB)	**Receiver Block Disable**
	Set to clear RXBLOCK, resulting in all incoming frames being loaded into the RX FIFO.			
4	RXBLOCKEN	0x0	W(nB)	**Receiver Block Enable**
	Set to set RXBLOCK, resulting in all incoming frames being discarded.			
3	TXDIS	0x0	W(nB)	**Transmitter Disable**
	Set to disable transmission. STATUS.TXENS should be polled to ensure disabled status in Synchrounous mode.			
2	TXEN	0x0	W(nB)	**Transmitter Enable**
	Set to enable data transmission. STATUS.TXENS should be polled to ensure enabled status in Synchrounous mode.			
1	RXDIS	0x0	W(nB)	**Receiver Disable**
	Set to disable data reception. If a frame is under reception when the receiver is disabled, the incoming frame is discarded. STATUS.RXENS should be polled to ensure disabled status in Synchrounous mode.			
0	RXEN	0x0	W(nB)	**Receiver Enable**
	Set to activate data reception. STATUS.RXENS should be polled to ensure enabled status in Synchrounous mode.			

21.5.16 EUSART_RXDATA - RX Data Register

Offset	Bit Position																													
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	R															
Name																	RXDATA															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	RXDATA	0x0	R	**RX Data and Control bits**
	Use this register to access data read from the EUSART, cleared on read access. In Synchronous mode, Bit 15:0 is actual RXDATA. In Asynchronous mode, if DALICFG.DALIEN is 0, then Bit 8:0 is actual RXDATA, bit 9 is PERR (set if received data has a parity error.), bit 10 is FERR (set if received data has a framing error, can be result of a break condition). In Asynchronous mode, if DALICFG.DALIEN is set to 1, then all 16 bits represent received data.			

21.5.17 EUSART_RXDATAP - RX Data Peek Register

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	R															
Name																	RXDATAP															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	RXDATAP	0x0	R	**RX Data Peek**
	Use this register to access data read from the EUSART without popping the FIFO.			

21.5.18 EUSART_TXDATA - TX Data Register

Offset	Bit Position																													
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	W															
Name																	TXDATA															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	TXDATA	0x0	W	**TX Data and Control bits** Use this register to write data to the EUSART. If TXEN is set, a transfer will be initiated at the first opportunity. In Synchronous mode, 15:0 is TXDATA. In Asynchronous mode, if DALICFG.DALIEN is set to 1, then all 16 bits represent data. In Asynchronous mode, if DALICFG.DALIEN is set to 0 then 8:0 is TXDATA, bit 9 is UBRXAT (Set to clear RXBLOCK after transmission, unblocking the receiver), bit 10 is TXTRIAT (Set to tristate transmitter by setting TXTRI after tranmission), bit 11 is TXBREAK (Set to send data as a break. Recipient will see a framing error or a break condition depending on its configuration and the value of TXDATA), bit 12 is TXDISAT (Set to disable trasmitter and release data bus directly after transmission), bit 13 is RXENAT (Set to enable reception after transmission).

21.5.19 EUSART_STATUS - Status Register

Offset	Bit Position																													
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0						0x0	0x0			0x0								0x1	0x1		0x0	0x0	0x0	0x0	0x1	0x0	0x0	0x0		0x0	0x0
Access	R						R	R			R								R	R		R	R	R	R	R	R	R	R		R	R
Name	RXFCNT						CLEARTXBUSY	AUTOBAUDDONE			TXFCNT								TXIDLE	RXIDLE		CLEARRXBUSY	TXCANDTXIDLE	RXFULL	RXFL	TXFL	TXC	TXTRI	RXBLOCK		TXENS	RXENS

Bit	Name	Reset	Access	Description
31:26	RXFCNT	0x0	R	**Valid entries in RX FIFO**
	Count of RX valid FIFO entries.			
25	CLEARTXBUSY	0x0	R	**TX FIFO Clear Busy**
	After issuing CLEARTX command, wait on this status flag until it goes low.			
24	AUTOBAUDDONE	0x0	R	**Auto Baud Rate Detection Completed**
	Set when auto baud rate has been detected and CLKDIV has been updated with required value. If AUTOBAUDEN is not set in CFG0 register, this bit is always read as '0'. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
23:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:16	TXFCNT	0x0	R	**Valid entries in TX FIFO**
	Count of TX valid FIFO entries.			
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	TXIDLE	0x1	R	**TX Idle**
	Set when TX idle. In Synchronous slave mode, TX is not considered idle when transmitting Default TX data.			
12	RXIDLE	0x1	R	**RX Idle**
	Set when RX is idle.			
11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	CLEARRXBUSY	0x0	R	**RX FIFO Clear Busy**
	After issuing CLEARRX command, wait on this status flag until it goes low.			
9	TXCANDTXIDLE	0x0	R	**And of TXC and TXIDLE**
	Set when both TXC and TXIDLE are set, indicating when TX is truly idle and done transmitting			
8	RXFULL	0x0	R	**RX FIFO Full**
	Set when the RX FIFO is full.			
7	RXFL	0x0	R	**RX FIFO Level**
	Set when data is available in the RX FIFO. Depends on the RXFIW setting in the CFG1 register.			
6	TXFL	0x1	R	**TX FIFO Level**
	Set when there is space for data in the TX FIFO. Depends on the TXFIW setting in CFG1 register.			
5	TXC	0x0	R	**TX Complete**
	Set when a transmission has completed and no more data is available in the TX FIFO and shift register.			
4	TXTRI	0x0	R	**Transmitter Tristated**

Bit	Name	Reset	Access	Description
	Set when the transmitter is tristated, and cleared when transmitter output is enabled. If AUTOTRI in UARTn_CFG is set, then this bit is always read as 0. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
3	RXBLOCK	0x0	R	**Block Incoming Data**
	When set, the receiver discards incoming frames. An incoming frame will not be loaded into the RX FIFO if this bit is set at the instant the frame has been completely received.			
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	TXENS	0x0	R	**Transmitter Enable Status**
	Set when the transmitter is enabled. In Synchronous slave mode, default TX data will be transmitted when the transmitter is disabled.			
0	RXENS	0x0	R	**Receiver Enable Status**
	Set when the receiver is enabled.			

21.5.20 EUSART_IF - Interrupt Flag Register

Offset	Bit Position																													
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset							0x0	0x0					0x0	0x0		0x0			0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access							RW	RW					RW	RW		RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name							RXTO	AUTOBAUDDONE					SIGF	STARTF		CSWU			TXIDLE	CCF	LOADERR	MPAF	FERR	PERR	TXUF	TXOF	RXUF	RXOF	RXFULL	RXFL	TXFL	TXC

Bit	Name	Reset	Access	Description
31:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	RXTO	0x0	RW	**RX Timeout Interrupt Flag**
	Set when RX timeout occurs. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
24	AUTOBAUDDONE	0x0	RW	**Auto Baud Complete Interrupt Flag**
	Set when auto baud rate detection is complete. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19	SIGF	0x0	RW	**Signal Frame Interrupt Flag**
	Set when a signal frame is detected. Please note that when MPA, START, and SIGNAL are set to match the same frame, corresponding interrupts might get triggered in arbitrary sequence due to synchronization uncertainty. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
18	STARTF	0x0	RW	**Start Frame Interrupt Flag**
	Set when a start frame is detected. Please note that when MPA, START, and SIGNAL are set to match the same frame, corresponding interrupts might get triggered in arbitrary sequence due to synchronization uncertainty. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	CSWU	0x0	RW	**CS Wake-up Interrupt Flag**
	Set when the CS asserts. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'SLAVE'.			
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	TXIDLE	0x0	RW	**TX Idle Interrupt Flag**
	Set when TX goes idle. In Synchronous mode, if TX is disabled during the middle of the transactions, TXIDLE won't be triggered when the engine becomes disabled.			
12	CCF	0x0	RW	**Collision Check Fail Interrupt Flag**
	Set when a collision check notices an error in the transmitted data. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
11	LOADERR	0x0	RW	**Load Error Interrupt Flag**
	Set when the empty TX FIFO is loaded less than the required TIMINGCFG.SETUPWINDOW bus clock cycles before the first edge of the incoming SCLK. Only applicable when CFG0.SYNC bit is set to 'SYNC' and CFG2.MASTER is set to 'SLAVE' and CFG2.FORCELOAD is set.			
10	MPAF	0x0	RW	**Multi-Processor Address Frame Interrupt**
	Set when a multi-processor address frame is detected. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			

Bit	Name	Reset	Access	Description
9	FERR	0x0	RW	**Framing Error Interrupt Flag**
	Set when a frame with a framing error is received while RXBLOCK is cleared. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
8	PERR	0x0	RW	**Parity Error Interrupt Flag**
	Set when a frame with a parity error is received while RXBLOCK is cleared. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
7	TXUF	0x0	RW	**TX FIFO Underflow Interrupt Flag**
	In Sync slave mode, Set when reading an empty TX FIFO with TX enabled. In Synchronous Master Mode with AUTOTX enabled, set when transmitting the word that does not contain valid data.			
6	TXOF	0x0	RW	**TX FIFO Overflow Interrupt Flag**
	Set when a write is done to the TX FIFO while it is full. The data already in the TX FIFO is preserved.			
5	RXUF	0x0	RW	**RX FIFO Underflow Interrupt Flag**
	Set when trying to read from the RX FIFO when it is empty.			
4	RXOF	0x0	RW	**RX FIFO Overflow Interrupt Flag**
	Set when data is complettly received in the receive shift register but the RX FIFO is full. RX FIFO is not overwirtten by new data.			
3	RXFULL	0x0	RW	**RX FIFO Full Interrupt Flag**
	Set when the RX FIFO becomes full.			
2	RXFL	0x0	RW	**RX FIFO Level Interrupt Flag**
	Set when data becomes available in the RX FIFO. This field depends on the RXFIW field in the CFG1 register.			
1	TXFL	0x0	RW	**TX FIFO Level Interrupt Flag**
	Set when space becomes available in the TX FIFO. This depends on the TXFIW field in the CFG1 register.			
0	TXC	0x0	RW	**TX Complete Interrupt Flag**
	This interrupt is set after a transmission when both the TX FIFO and shift register are empty.			

21.5.21 EUSART_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset							0x0	0x0					0x0	0x0		0x0			0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access							RW	RW					RW	RW		RW			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name							RXTO	AUTOBAUDDONE					SIGF	STARTF		CSWU			TXIDLE	CCF	LOADERR	MPAF	FERR	PERR	TXUF	TXOF	RXUF	RXOF	RXFULL	RXFL	TXFL	TXC

Bit	Name	Reset	Access	Description
31:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	RXTO	0x0	RW	**RX Timeout**
	Interrupt enable for RXTOIF.			
24	AUTOBAUDDONE	0x0	RW	**Auto Baud Complete**
	Interrupt enable for AUTOBAUDDONEIF.			
23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19	SIGF	0x0	RW	**Signal Frame**
	Interrupt enable for SIGFIF.			
18	STARTF	0x0	RW	**Start Frame**
	Interrupt enable for STARTFIF.			
17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	CSWU	0x0	RW	**CS Wake-up**
	Interrupt enable for CSWUIF.			
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	TXIDLE	0x0	RW	**TX IDLE**
	Interrupt enable for TXIDLEIF.			
12	CCF	0x0	RW	**Collision Check Fail**
	Interrupt enable for CCFIF.			
11	LOADERR	0x0	RW	**Load Error**
	Interrupt enable for LOADERRIF.			
10	MPAF	0x0	RW	**Multi-Processor Addr Frame**
	Interrupt enable for MPAFIF.			
9	FERR	0x0	RW	**Framing Error**
	Interrupt enable for FERRIF.			
8	PERR	0x0	RW	**Parity Error**
	Interrupt enable for PERRIF.			
7	TXUF	0x0	RW	**TX FIFO Underflow**

Bit	Name	Reset	Access	Description
	Interrupt enable for TXUFIF.			
6	TXOF	0x0	RW	**TX FIFO Overflow**
	Interrupt enable for TXOFIF.			
5	RXUF	0x0	RW	**RX FIFO Underflow**
	Interrupt enable for RXUFIF.			
4	RXOF	0x0	RW	**RX FIFO Overflow**
	Interrupt enable for RXOFIF.			
3	RXFULL	0x0	RW	**RX FIFO Full**
	Interrupt enable for RXFULLIF.			
2	RXFL	0x0	RW	**RX FIFO Level**
	Interrupt enable for RXFLIF.			
1	TXFL	0x0	RW	**TX FIFO Level**
	Interrupt enable for TXFLIF.			
0	TXC	0x0	RW	**TX Complete**
	Interrupt enable for TXCIF.			

21.5.22 EUSART_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																													
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																					R	R	R	R	R	R	R	R	R	R	R	R
Name																					AUTOTXTEN	TXTRIDIS	TXTRIEN	RXBLOCKDIS	RXBLOCKEN	TXDIS	TXEN	RXDIS	RXEN	TXTEN	RXTEN	DIV

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	AUTOTXTEN	0x0	R	**SYNCBUSY for AUTOTXTEN in TRIGCTRL**
	This bit is set when there is an ongoing synchronization of AUTOTXTEN field. Do not do another write to the same field while this bit is set.			
10	TXTRIDIS	0x0	R	**SYNCBUSY in TXTRIDIS in CMD**
	This bit is set when there is an ongoing synchronization of TXTRIDIS field. Do not do another write to the same field while this bit is set. Only applicable when CFG0.SYNC bit is set to 'ASYNC'.			
9	TXTRIEN	0x0	R	**SYNCBUSY for TXTRIEN in CMD**
	This bit is set when there is an ongoing synchronization of TXTRIEN field. Do not do another write to the same field while this bit is set.			
8	RXBLOCKDIS	0x0	R	**SYNCBUSY for RXBLOCKDIS in CMD**
	This bit is set when there is an ongoing synchronization of RXBLOCKDIS field. Do not do another write to the same field while this bit is set.			
7	RXBLOCKEN	0x0	R	**SYNCBUSY for RXBLOCKEN in CMD**
	This bit is set when there is an ongoing synchronization of RXBLOCKEN field. Do not do another write to the same field while this bit is set.			
6	TXDIS	0x0	R	**SYNCBUSY for TXDIS in CMD**
	This bit is set when there is an ongoing synchronization of TXDIS field. Do not do another write to the same field while this bit is set.			
5	TXEN	0x0	R	**SYNCBUSY for TXEN in CMD**
	This bit is set when there is an ongoing synchronization of TXEN field. Do not do another write to the same field while this bit is set.			
4	RXDIS	0x0	R	**SYNCBUSY for RXDIS in CMD**
	This bit is set when there is an ongoing synchronization of RXDIS field. Do not do another write to the same field while this bit is set.			
3	RXEN	0x0	R	**SYNCBUSY for RXEN in CMD**
	This bit is set when there is an ongoing synchronization of RXEN field. Do not do another write to the same field while this bit is set.			
2	TXTEN	0x0	R	**SYNCBUSY for TXTEN in TRIGCTRL**
	This bit is set when there is an ongoing synchronization of TXTEN field. Do not do another write to the same field while this bit is set.			
1	RXTEN	0x0	R	**SYNCBUSY for RXTEN in TRIGCTRL**
	This bit is set when there is an ongoing synchronization of RXTEN field. Do not do another write to the same field while this bit is set.			

Bit	Name	Reset	Access	Description
0	DIV	0x0	R	**SYNCBUSY for DIV in CLKDIV** This bit is set when there is an ongoing synchronization of DIV field. Do not do another write to the same field while this bit is set.

21.5.23 EUSART_DALICFG - DALI Config Register

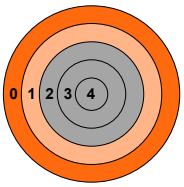
Offset	Bit Position																													
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0			0x0							0x0					0x0
Access																	RW			RW							RW					RW
Name																	DALIRXENDT			DALIRXDATABITS							DALITXDATABITS					DALIEN

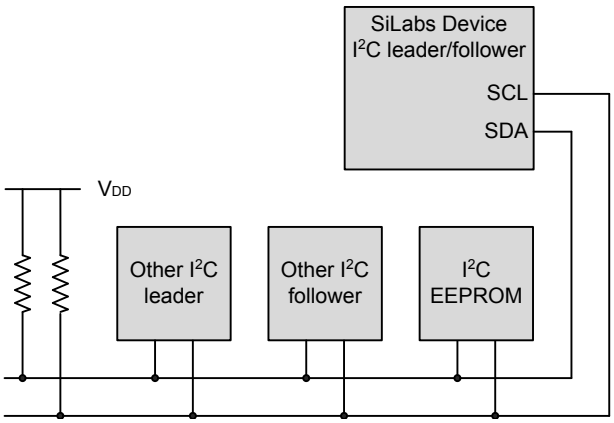
Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15	DALIRXENDT	0x0	RW	**DALI RX Enabled During Transmission**
	If set, then EUSART RX will stay enabled during EUSART DALI TX. Should be set in loopback mode.			
14:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12:8	DALIRXDATABITS	0x0	RW	**DALI RX Databits**
	DALI RX databits (8-32). Only used if DALIEN is set.			

Value	Mode	Description
0	EIGHT	Each frame contains 8 data bits
1	NINE	Each frame contains 9 data bits
2	TEN	Each frame contains 10 data bits
3	ELEVEN	Each frame contains 11 data bits
4	TWELVE	Each frame contains 12 data bits
5	THIRTEEN	Each frame contains 13 data bits
6	FOURTEEN	Each frame contains 14 data bits
7	FIFTEEN	Each frame contains 15 data bits
8	SIXTEEN	Each frame contains 16 data bits
9	SEVENTEEN	Each frame contains 17 data bits
10	EIGHTEEN	Each frame contains 18 data bits
11	NINETEEN	Each frame contains 19 data bits
12	TWENTY	Each frame contains 20 data bits
13	TWENTYONE	Each frame contains 21 data bits
14	TWENTYTWO	Each frame contains 22 data bits
15	TWENTYTHREE	Each frame contains 23 data bits
16	TWENTYFOUR	Each frame contains 24 data bits
17	TWENTYFIVE	Each frame contains 25 data bits
18	TWENTYSIX	Each frame contains 26 data bits
19	TWENTYSEVEN	Each frame contains 27 data bits
20	TWENTYEIGHT	Each frame contains 28 data bits

Bit	Name	Reset	Access	Description
	21	TWENTYNINE		Each frame contains 29 data bits
	22	THIRTY		Each frame contains 30 data bits
	23	THIRTYONE		Each frame contains 31 data bits
	24	THIRTYTWO		Each frame contains 32 data bits
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:1	DALITXDATABITS	0x0	RW	**DALI TX Databits**
	DALI TX databits (8-32). Only used if DALIEN is set.			
	Value	Mode		Description
	0	EIGHT		Each frame contains 8 data bits
	1	NINE		Each frame contains 9 data bits
	2	TEN		Each frame contains 10 data bits
	3	ELEVEN		Each frame contains 11 data bits
	4	TWELVE		Each frame contains 12 data bits
	5	THIRTEEN		Each frame contains 13 data bits
	6	FOURTEEN		Each frame contains 14 data bits
	7	FIFTEEN		Each frame contains 15 data bits
	8	SIXTEEN		Each frame contains 16 data bits
	9	SEVENTEEN		Each frame contains 17 data bits
	10	EIGHTEEN		Each frame contains 18 data bits
	11	NINETEEN		Each frame contains 19 data bits
	12	TWENTY		Each frame contains 20 data bits
	13	TWENTYONE		Each frame contains 21 data bits
	14	TWENTYTWO		Each frame contains 22 data bits
	15	TWENTYTHREE		Each frame contains 23 data bits
	16	TWENTYFOUR		Each frame contains 24 data bits
	17	TWENTYFIVE		Each frame contains 25 data bits
	18	TWENTYSIX		Each frame contains 26 data bits
	19	TWENTYSEVEN		Each frame contains 27 data bits
	20	TWENTYEIGHT		Each frame contains 28 data bits
	21	TWENTYNINE		Each frame contains 29 data bits
	22	THIRTY		Each frame contains 30 data bits
	23	THIRTYONE		Each frame contains 31 data bits
	24	THIRTYTWO		Each frame contains 32 data bits
0	DALIEN	0x0	RW	**DALI Enable Bit**
	DALI EN bit.			

22. I2C - Inter-Integrated Circuit Interface





Quick Facts

What?

The I²C interface allows communication on I²C-buses with the lowest energy consumption possible.

Why?

I²C is a popular serial bus that enables communication with a number of external devices using only two I/O pins.

How?

With the help of DMA, the I²C interface allows I²C communication with minimal CPU intervention. Address recognition is available in all energy modes (except EM4), allowing the MCU to wait for data on the I²C-bus with sub-µA current consumption.

22.1 Introduction

The I²C module provides an interface between the MCU and a serial I²C-bus. It is capable of acting as both a leader and a follower and supports multi-leader buses. Standard-mode, fast-mode and fast-mode plus speeds are supported, allowing transmission rates all the way from 10 kbit/s up to 1 Mbit/s. Follower arbitration and timeouts are also provided to allow implementation of an SMBus compliant system. The interface provided to software by the I²C module allows precise control of the transmission process and highly automated transfers. Automatic recognition of follower addresses is provided in all energy modes (except EM4).

22.2 Features

- True multi-leader capability
- Support for different bus speeds
  - Standard-mode (Sm) bit rate up to 100 kbit/s
  - Fast-mode (Fm) bit rate up to 400 kbit/s
  - Fast-mode Plus (Fm+) bit rate up to 1 Mbit/s
- Arbitration for both leader and follower (allows SMBus ARP)
- Clock synchronization and clock stretching
- Hardware address recognition
  - 7-bit masked address
  - General call address
- 10-bit address support
- Error handling
  - Clock low timeout
  - Bus idle (clock high) timeout
  - Arbitration lost
  - Bus error detection
- Separate receive/ transmit 2-deep FIFOs, with additional separate shift registers
- Full DMA support

22.3 Functional Description

An overview of the I2C module is shown in Figure 22.1. I2C Overview.

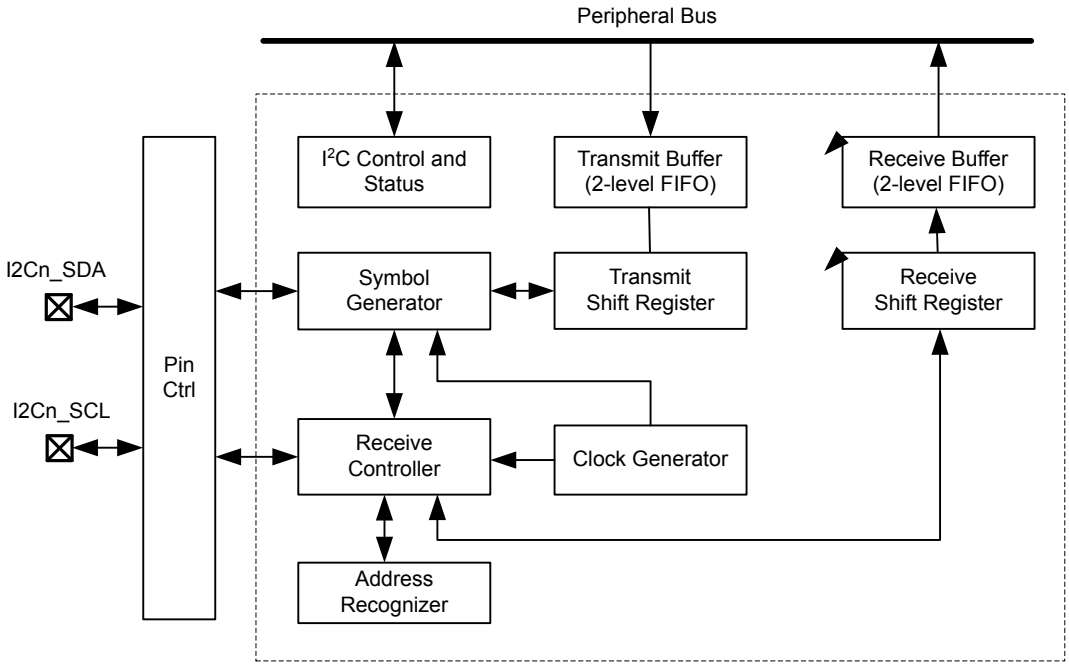


Figure 22.1. I2C Overview

22.3.1 Power Domains and Energy Modes

Individual I2C instances are powered from different power domains in the device, which may impact clocking options and availability in different energy modes. See 8. CMU - Clock Management Unit for more details on the clocking. The power domain and energy mode availability for each instance are summarized in the following table.

Table 22.1. I2C Power Domains

Instance	Power Domain	Energy Modes
I2C0	PD0D	EM0, EM1
I2C1	PD1	EM0, EM1
I2C2	PD1	EM0, EM1

22.3.2 I2C-Bus Overview

The I^2C-bus uses two wires for communication; a serial data line (SDA) and a serial clock line (SCL) as shown in Figure 22.2. I2C-Bus Example. As a true multi-leader bus it includes collision detection and arbitration to resolve situations where multiple leaders transmit data at the same time without data loss.

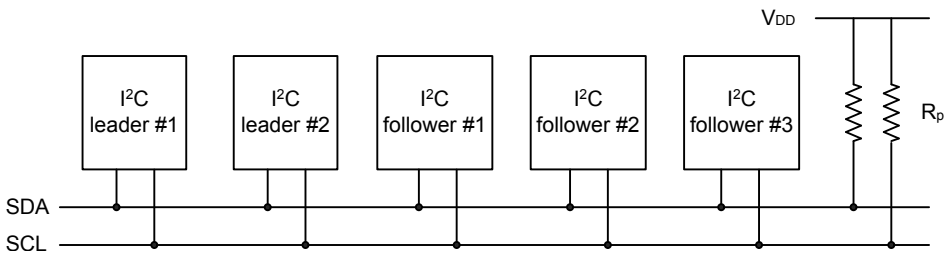


Figure 22.2. I2C-Bus Example

Each device on the bus is addressable by a unique address, and an I^2C leader can address all the devices on the bus, including other leaders.

Both the bus lines are open-drain. The maximum value of the pull-up resistor can be calculated as a function of the maximal rise-time **tr** for the given bus speed, and the estimated bus capacitance **Cb** as shown in Figure 22.3. I2C Pull-up Resistor Equation.

$$R_{p(max)} = t_r/(0.8473 \times C_b).$$

Figure 22.3. I2C Pull-up Resistor Equation

The maximal rise times for 100 kHz, 400 kHz and 1 MHz I^2C are 1 µs, 300 ns and 120 ns respectively.

Note: The GPIO slew rate control should be set for the desired slew rate..

Note: If V_{dd} drops below the voltage on SCL and SDA lines, the MCU could become back powered and pull the SCL and SDA lines low.

22.3.2.1 START and STOP Conditions

START and STOP conditions are used to initiate and stop transactions on the I²C-bus. All transactions on the bus begin with a START condition (S) and end with a STOP condition (P). As shown in Figure 22.4. I2C START and STOP Conditions, a START condition is generated by pulling the SDA line low while SCL is high, and a STOP condition is generated by pulling the SDA line high while SCL is high.

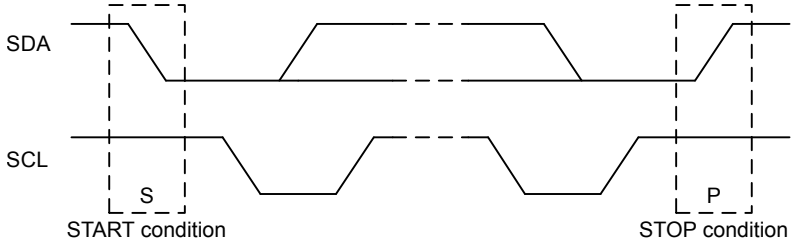


Figure 22.4. I2C START and STOP Conditions

The START and STOP conditions are easily identifiable bus events as they are the only conditions on the bus where a transition is allowed on SDA while SCL is high. During the actual data transmission, SDA is only allowed to change while SCL is low, and must be stable while SCL is high. One bit is transferred per clock pulse on the I²C-bus as shown in Figure 22.5. I2C Bit Transfer on I²C-Bus.

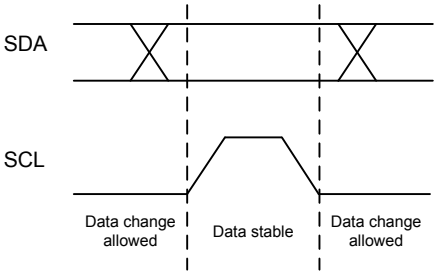


Figure 22.5. I2C Bit Transfer on I²C-Bus

22.3.2.2 Bus Transfer

When a leader wants to initiate a transfer on the bus, it waits until the bus is idle and transmits a START condition on the bus. The leader then transmits the address of the follower it wishes to interact with and a single R/W bit telling whether it wishes to read from the follower (R/W bit set to 1) or write to the follower (R/W bit set to 0).

After the 7-bit address and the R/W bit, the leader releases the bus, allowing the follower to acknowledge the request. During the next bit-period, the follower pulls SDA low (ACK) if it acknowledges the request, or keeps it high if it does not acknowledge it (NACK).

Following the address acknowledge, either the follower or leader transmits data, depending on the value of the R/W bit. After every 8 bits (one byte) transmitted on the SDA line, the transmitter releases the line to allow the receiver to transmit an ACK or a NACK. Both the data and the address are transmitted with the most significant bit first.

The number of bytes in a bus transfer is unrestricted. The leader ends the transmission after a (N)ACK by sending a STOP condition on the bus. After a STOP condition, any leader wishing to initiate a transfer on the bus can try to gain control of it. If the current leader wishes to make another transfer immediately after the current, it can start a new transfer directly by transmitting a repeated START condition (Sr) instead of a STOP followed by a START.

Examples of I^2C transfers are shown in Figure 22.6. I2C Single Byte Write to Follower, and Figure 22.7. I2C Single Byte Write, then Repeated Start and Single Byte Read. The identifiers used are:

- ADDR - Address
- DATA - Data
- S - Start bit
- Sr - Repeated start bit
- P - Stop bit
- W/R - Read(1)/Write(0)
- A - ACK
- N - NACK

Figure 22.6. I2C Single Byte Write to Follower

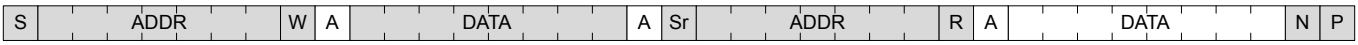


Figure 22.7. I2C Single Byte Write, then Repeated Start and Single Byte Read

22.3.2.3 Addresses

I²C supports both 7-bit and 10-bit addresses. When using 7-bit addresses, the first byte transmitted after the START-condition contains the address of the follower that the leader wants to contact. In the 7-bit address space, several addresses are reserved. These addresses are summarized in Table 22.2. I2C Reserved I²C Addresses, and include a General Call address which can be used to broadcast a message to all followers on the I²C-bus.

Table 22.2. I2C Reserved I²C Addresses

I²C Address	R/W	Description
0000-000	0	General Call address
0000-000	1	START byte
0000-001	X	Reserved for the C-Bus format
0000-010	X	Reserved for a different bus format
0000-011	X	Reserved for future purposes
0000-1XX	X	Reserved for future purposes
1111-1XX	X	Reserved for future purposes
1111-0XX	X	10 Bit follower addressing mode

22.3.2.4 10-bit Addressing

To address a follower using a 10-bit address, two bytes are required to specify the address instead of one. The seven first bits of the first byte must then be 1111 0XX, where XX are the two most significant bits of the 10-bit address. As with 7-bit addresses, the eighth bit of the first byte determines whether the leader wishes to read from or write to the follower. The second byte contains the eight least significant bits of the follower address.

When a follower receives a 10-bit address, it must acknowledge both the address bytes if they match the address of the follower.

When performing a leader transmitter operation, the leader transmits the two address bytes and then the remaining data, as shown in Figure 22.8. I2C Leader Transmitter/Follower Receiver with 10-bit Address.

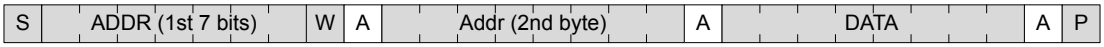


Figure 22.8. I2C Leader Transmitter/Follower Receiver with 10-bit Address

When performing a leader receiver operation however, the leader first transmits the two address bytes in a leader transmitter operation, then sends a repeated START followed by the first address byte and then receives data from the addressed follower. The follower addressed by the 10-bit address in the first two address bytes must remember that it was addressed, and respond with data if the address transmitted after the repeated start matches its own address. An example of this (with one byte transmitted) is shown in Figure 22.9. I2C Leader Receiver/Follower Transmitter with 10-bit Address.

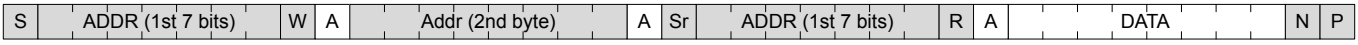


Figure 22.9. I2C Leader Receiver/Follower Transmitter with 10-bit Address

22.3.2.5 Arbitration, Clock Synchronization, Clock Stretching

Arbitration and clock synchronization are features aimed at allowing multi-leader buses. Arbitration occurs when two devices try to drive the bus at the same time. If one device drives it low, while the other drives it high, the one attempting to drive it high will not be able to do so due to the open-drain bus configuration. Both devices sample the bus, and the one that was unable to drive the bus in the desired direction detects the collision and backs off, letting the other device continue communication on the bus undisturbed.

Clock synchronization is a means of synchronizing the clock outputs from several leaders driving the bus at once, and is a requirement for effective arbitration.

Followers on the bus are allowed to force the clock output on the bus low in order to pause the communication on the bus and give themselves time to process data or perform any real-time tasks they might have. This is called clock stretching.

Arbitration is supported by the I^2C module for both leaders and followers. Clock synchronization and clock stretching is also supported.

22.3.3 Enable and Reset

The I^2C is enabled by setting the EN bit in the I2C_EN register.

To reset the internal state of the I^2C module and terminate any ongoing transfers, set the CORERST bit in I2C_CTRL. After resetting, the CORERST bit must be cleared to resume I^2C operation.

Note: When enabling the I^2C, the ABORT command or the Bus Idle Timeout feature must be applied prior to use even if the BUSY flag is not set.

22.3.4 Pin Configuration

The I^2C SDA and SCL pins are configured and enabled in the GPIO_I2Cn_ROUTEEN, GPIO_I2Cn_SCLROUTE, and GPIO_I2Cn_SDAROUTE registers.

22.3.5 Safely Disabling and Changing Follower Configuration

The I^2C follower is partially asynchronous, and some precautions are necessary to always ensure a safe follower disable or follower configuration change. These measures should be taken, if (while the follower is enabled) the user cannot guarantee that an address match will not occur at the exact time of follower disable or follower configuration change.

Worst case consequences for an address match while disabling follower or changing configuration is that the follower may end up in an undefined state. To reset the follower back to a known state, the EN bit in I2C_EN must be cleared. This should be done regardless of whether the follower is going to be re-enabled or not.

22.3.6 Clock Generation

The I^2C peripheral clock (I2CCLK) is derived from the EM01GRPDCLK.

The SCL signal generated by the I^2C leader determines the maximum transmission rate on the bus. The clock is generated as a division of the peripheral clock (I2CCLK), and is given by the following equation:

$$f_{SCL} = f_{I2CCLK}/(((N_{low} + N_{high}) \times (DIV + 1)) + 8 + N_{fall} + N_{rise})$$

Figure 22.10. I2C Maximum Transmission Rate

Where DIV is the clock divider value set in I2C_CLKDIV, the values of N_{low} and N_{high} (and thus the ratio between the high and low parts of the clock signal) are controlled by CLHR in the I2C_CTRL register, and N_{fall} and N_{rise} represent the number of I2CCLK cycles required for clock synchronization.

The values of N_{low} and N_{high} specify the number of prescaled clock cycles in the low and high periods of the clock signal respectively. The worst case low and high periods of the signal are:

$$t_{high} >= (N_{high} \times (DIV + 1) + 4)/f_{I2CCLK}$$

$$t_{low} >= (N_{low} \times (DIV + 1) + 4)/f_{I2CCLK}$$

Figure 22.11. I2C High and Low Cycles Equations

Clock synchronization is used to ensure that requested low and high times are met on the bus. The counters establishing high and low time are only active once the pin logic has reached the high or low logic levels, and so the rise and fall times will impact the maximum transmission rate on the bus. The clock logic level is sampled at a rate of f_{I2CCLK}, and will therefore be quantized to an integer number of I2CCLK clock cycles, as:

$$N_{rise} = CEILING(t_{rise}/t_{I2CCLK})$$

$$N_{fall} = CEILING(t_{fall}/t_{I2CCLK})$$

Figure 22.12. I2C Clock Synchronization Cycles

Note that there is an inherent propagation delay between driving SCL and sampling the signal, so the above equations can result in N = 0 for fast rise or fall times.

For example, a system with a weak pull-up on SCL may result in long t_{rise}, requiring several N_{rise} synchronization cycles, and subsequently impact the I2C transmission rate. In the same system, t_{fall} may be faster than the sampling delay, resulting in N_{fall} = 0.

Note: DIV must be set to 1 during follower mode operation.

22.3.7 Arbitration

Arbitration is enabled by default, but can be disabled by setting the ARBDIS bit in I2C_CTRL. When arbitration is enabled, the value on SDA is sensed every time the I^2C module attempts to change its value. If the sensed value is different than the value the I^2C module tried to output, it is interpreted as a simultaneous transmission by another device, and that the I^2C module has lost arbitration.

Whenever arbitration is lost, the ARBLOST interrupt flag in I2C_IF is set, any lines held are released, and the I^2C device goes idle. If an I^2C leader loses arbitration during the transmission of an address, another leader may be trying to address it. The leader therefore receives the rest of the address, and if the address matches the follower address of the leader, the leader goes into either follower transmitter or follower receiver mode.

Note:

Arbitration can be lost both when operating as a leader and when operating as a follower.

22.3.8 FIFOs

22.3.8.1 Transmit Buffer and Shift Register

The I²C transmitter has a 2-level FIFO transmit buffer and a transmit shift register as shown in Figure 22.1. I2C Overview. A byte is loaded into the transmit buffer by writing to I2C_TXDATA. Figure 22.13. I2C Transmit Buffer Operation shows the basics of the transmit buffer. When the transmit shift register is empty and ready for new data, the byte from the transmit buffer is then loaded into the shift register. The byte is then kept in the shift register until it is transmitted. When a byte has been transmitted, a new byte is loaded into the shift register (if available in the transmit buffer). If the transmit buffer is empty, then the shift register also remains empty. The TXC flag in I2C_STATUS and the TXC interrupt flags in I2C_IF are then set, signaling that the transmit shift register is out of data. TXC is cleared when new data becomes available, but the TXC interrupt flag must be cleared by software.

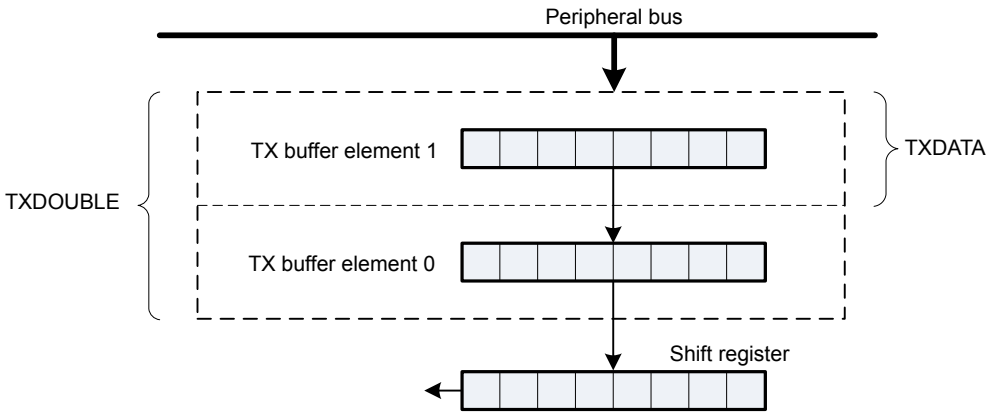


Figure 22.13. I2C Transmit Buffer Operation

In addition to the interrupt flag TXCIF in I2Cn_IF and status flag TXC in I2Cn_STATUS which are set when the transmission is complete, TXFL in I2Cn_STATUS and the TXFLIF interrupt flag in I2Cn_IF are used to indicate the level of the transmit FIFO. TXFIL in I2Cn_CTRL controls the level at which these bits are set.

TXFL and TXFLIF Indicates the level of the transmit fifo. If TXFIL=0, then both the status and interrupt flags are set when the TX FIFO has space for at least one more frame. If TXFIL=1, then both the status and interrupt flags are set when the TX FIFO has space for at least two more frames. Both the TXFL status flag and the TXFLIF interrupt flag are cleared automatically when the condition becomes false.

Additionally, the TXFCNT bitfield in I2C_STATUS can be read to determine the exact number of bytes in the transmit FIFO. This is particularly useful for determining whether the transmit buffers are full. For example, if TXFCNT = '2', firmware can determine that the buffer is full, and that any additional data written to the transmit buffer would result in an overflow condition. Note that the TXFCNT count does not include the TX shift register.

If an attempt is made to write more bytes to the transmit buffer than the space available, the TXOF interrupt flag in I2C_IF is set, indicating the overflow. The data already in the buffer remains preserved, and no new data is written.

The transmit buffer and the transmit shift register can be cleared by setting command bit CLEARTX in I2C_CMD. This will prevent the I²C module from transmitting the data in the buffer and the shift register, and will make them available for new data. Any byte currently being transmitted will not be aborted. Transmission of this byte will be completed.

22.3.8.2 Receive Buffer and Shift Register

The I²C receiver uses a 2-level FIFO receive buffer and a receive shift register as shown in Figure 22.14. I2C Receive Buffer Operation. When a byte has been fully received by the receive shift register, it is loaded into the receive buffer if there is room for it, making the shift register empty to receive another byte. Otherwise, the byte waits in the shift register until space becomes available in the buffer.

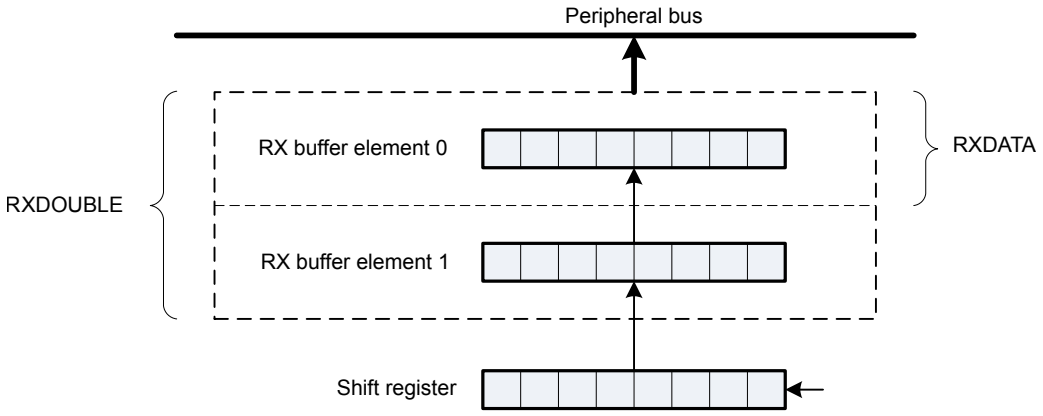


Figure 22.14. I2C Receive Buffer Operation

When a byte becomes available in the receive buffer, the RXDATAV flags in I2C_STATUS and I2C_IF are set. When the buffer becomes full, the RXFULL flags in I2C_STATUS and I2C_IF are set. The RXDATAV and RXFULL flags in I2C_STATUS are automatically cleared by hardware when their condition is no longer true. The RXDATAV and RXFULL flags in I2C_IF must be manually cleared by software after the receive FIFO is emptied. Note that when the RXFULL flag is set, indicating the buffer is full, space is still available in the receive shift register for one more byte.

The data can be fetched from the buffer by reading I2C_RXDATA, which gives access to the received byte (if two bytes are received then the one received first is fetched first). If an attempt is made to read more bytes from the buffer than available, the RXUF interrupt flag in I2C_IF is set to signal the underflow, and the data read from the buffer is undefined.

I2C_RXDATAP can be used to read data from the receive buffer without removing it from the buffer. The RXUF interrupt flag in I2C_IF will never be set as a result of reading from I2C_RXDATAP, but the data read through I2C_RXDATAP when the receive buffer is empty is still undefined.

Once a transaction is complete (STOP sent or received), the receive fifo needs to be flushed (all received data must be read) before starting a new transaction.The recieve fifo and the recieve shift register can be cleared by setting command bit CLEARRX in I2C_CMD.

22.3.9 Leader Operation

A bus transaction is initiated by transmitting a START condition (S) on the bus. This is done by setting the START bit in I2C_CMD. The command schedules a START condition, and makes the I²C module generate a start condition whenever the bus becomes free.

The I²C-bus is considered busy whenever another device on the bus transmits a START condition. Until a STOP condition is detected, the bus is owned by the leader issuing the START condition. The bus is considered free when a STOP condition is transmitted on the bus. After a STOP is detected, all leaders that have data to transmit send a START condition and begin transmitting data. Arbitration ensures that collisions are avoided.

When the START condition has been transmitted, the leader must transmit a follower address (ADDR) with an R/W bit on the bus. If this address is available in the transmit buffer, the leader transmits it immediately, but if the buffer is empty, the leader holds the I²C-bus while waiting for software to write the address to the transmit buffer.

After the address has been transmitted, a sequence of bytes can be read from or written to the follower, depending on the value of the R/W bit (bit 0 in the address byte). If the bit was cleared, the leader has entered a leader transmitter role, where it now transmits data to the follower. If the bit was set, it has entered a leader receiver role, where it now should receive data from the follower. In either case, an unlimited number of bytes can be transferred in one direction during the transmission.

At the end of the transmission, the leader either transmits a repeated START condition (Sr) if it wishes to continue with another transfer, or transmits a STOP condition (P) if it wishes to release the bus. When operating in the leader mode, I2CCLK frequency must be higher than 2 MHz for Standard-mode, 9 MHz for Fast-mode, and 20 MHz for Fast-mode Plus.

22.3.9.1 Leader State Machine

The leader state machine is shown in Figure 22.15. I2C Leader State Machine. A leader operation starts in the far left of the state machine, and follows the solid lines through the state machine, ending the operation or continuing with a new operation when arriving at the right side of the state machine.

Branches in the path through the state machine are the results of bus events and choices made by software, either directly or indirectly. The dotted lines show where I²C-specific interrupt flags are set along the path and the full-drawn circles show places where interaction may be required by software to let the transmission proceed.

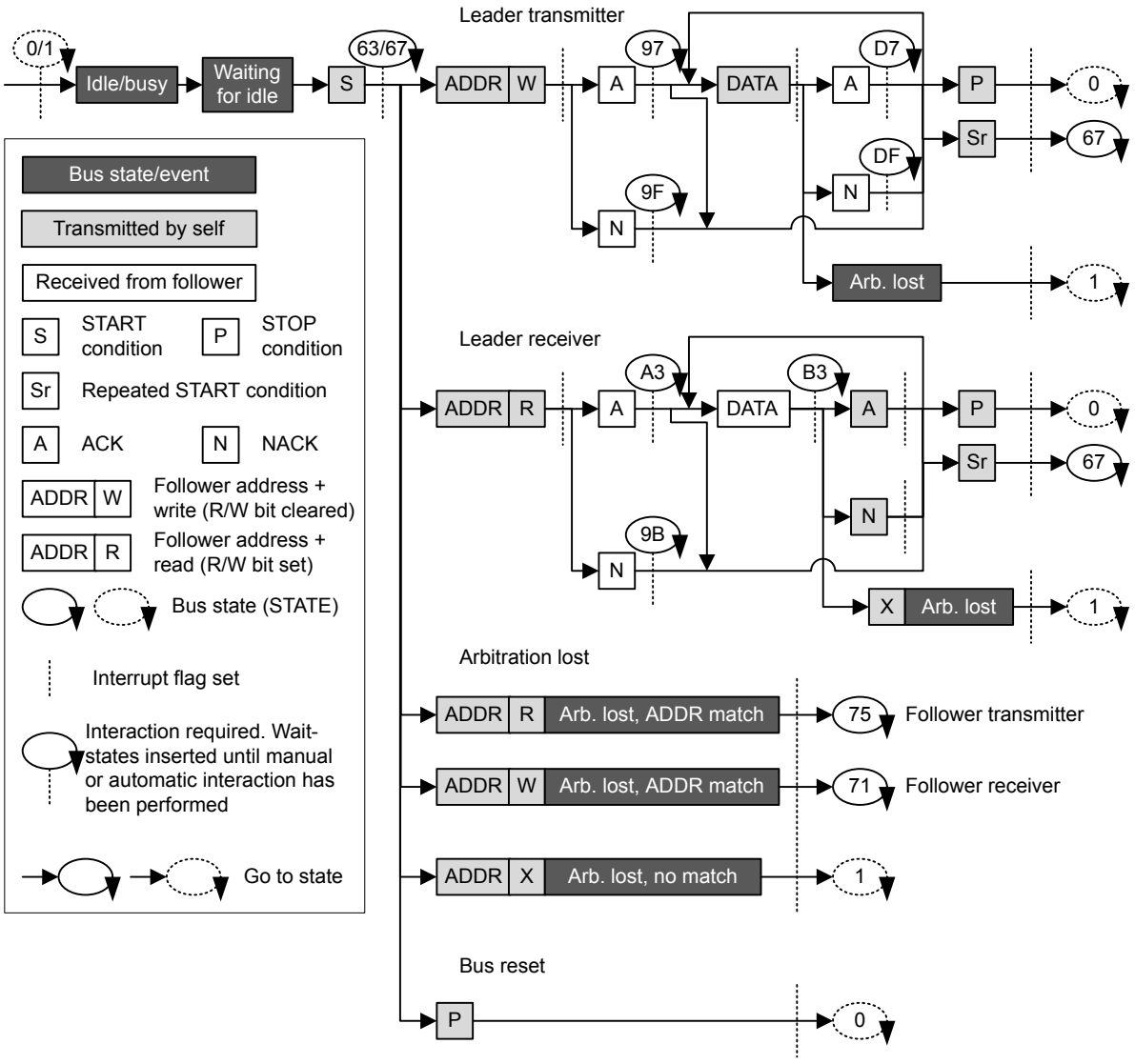


Figure 22.15. I2C Leader State Machine

22.3.9.2 Interactions

Whenever the I²C module is waiting for interaction from software, it holds the bus clock SCL low, freezing all bus activities, and the BUSHOLD interrupt flag in I2C_IF is set. The action(s) required by software depends on the current state the of the I²C module. This state can be read from the I2C_STATE register.

As an example, Table 22.4. I2C Leader Transmitter shows the different states the I²C goes through when operating as a Leader Transmitter, i.e., a leader that transmits data to a follower. As seen in the table, when a start condition has been transmitted, a requirement is that there is an address and an R/W bit in the transmit buffer. If the transmit buffer is empty, then the BUSHOLD interrupt flag is set, and the bus is held until data becomes available in the buffer. While waiting for the address, I2C_STATE has a value 0x67, which can be used to identify exactly what the I²C module is waiting for.

Note: The bus would never stop at state 0x67 if the address was available in the transmit buffer.

The BUSHOLD interrupt flag needs to be manually cleared by software after the appropriate action has been taken.

The different interactions used by the I²C module are listed in Table 22.3. I2C Interactions in Prioritized Order in a prioritized order. If the I²C module is in such a state that multiple courses of action are possible, then the action chosen is the one that has the highest priority. For example, after sending out a START, if an address is present in the buffer and a STOP is also pending, then the I²C will send out the STOP since it has the higher priority.

Table 22.3. I2C Interactions in Prioritized Order

Interaction	Priority	Software action	Automatically continues if
STOP*	1	Set the STOP command bit in I2C_CMD	PSTOP is set (STOP pending) in I2C_STATUS
ABORT	2	Set the ABORT command bit in I2C_CMD	Never, the transmission is aborted
CONT*	3	Set the CONT command bit in I2C_CMD	PCONT is set in I2C_STATUS (CONT pending)
NACK*	4	Set the NACK command bit in I2C_CMD	PNACK is set in I2C_STATUS (NACK pending)
ACK*	5	Set the ACK command bit in I2C_CMD	AUTOACK is set in I2C_CTRL or PACK is set in I2C_STATUS (ACK pending)
ADDR+W -> TXDATA	6	Write an address to the transmit buffer with the R/W bit set	Address is available in transmit buffer with R/W bit set
ADDR+R -> TXDATA	7	Write an address to the transmit buffer with the R/W bit cleared	Address is available in transmit buffer with R/W bit cleared
START*	8	Set the START command bit in I2C_CMD	PSTART is set in I2C_STATUS (START pending)
TXDATA	9	Write data to the transmit buffer	Data is available in transmit buffer
RXDATA	10	Read data from receive buffer	Space is available in receive buffer
None	11	No interaction is required	

The commands marked with a * in Table 22.3. I2C Interactions in Prioritized Order can be issued before an interaction is required. When such a command is issued before it can be used/consumed by the I²C module, the command is set in a pending state, which can be read from the STATUS register. A pending START command can for instance be identified by PSTART having a high value.

Whenever the I²C module requires an interaction, it checks the pending commands. If one or a combination of these can fulfill an interaction, they are consumed by the module and the transmission continues without setting the BUSHOLD interrupt flag in I2C_IF to get an interaction from software. The pending status of a command goes low when it is consumed.

When several interactions are possible from a set of pending commands, the interaction with the highest priority, i.e., the interaction closest to the top of Table 22.3. I2C Interactions in Prioritized Order is applied to the bus.

Pending commands can be cleared by setting the CLEARPC command bit in I2C_CMD.

22.3.9.3 Automatic ACK Interaction

When receiving addresses and data, an ACK command in I2C_CMD is normally required after each received byte. When AUTOACK is set in I2C_CTRL, an ACK is always pending, and the ACK-pending bit PACK in I2C_STATUS is thus always set, even after an ACK has been consumed. This can be used with I2C_RXDATA in order to reduce the amount of software interaction required during a transfer.

22.3.9.4 Reset State

After a reset, the state of the I^2C-bus is unknown. To avoid interrupting transfers on the I^2C-bus after a reset of the I^2C module or the entire MCU, the I^2C-bus is assumed to be busy when coming out of a reset, and the BUSY flag in I2C_STATUS is thus set. To be able to carry through leader operations on the I^2C-bus, the bus must be idle.

The bus goes idle when a STOP condition is detected on the bus, but on buses with little activity, the time before the I^2C module detects that the bus is idle can be significant. There are two ways of assuring that the I^2C module gets out of the busy state.

- Use the ABORT command in I2C_CMD. When the ABORT command is issued, the I^2C module is instructed that the bus is idle. The I^2C module can then initiate leader operations.
- Use the Bus Idle Timeout. When SCL has been high for a long period of time, it is very likely that the bus is idle. Set BITO in I2C_CTRL to an appropriate timeout period and set GIBITO in I2C_CTRL. If activity has not been detected on the bus within the timeout period, the bus is then automatically assumed idle, and leader operations can be initiated.

Note: If operating in follower mode, the above approach is not necessary.

22.3.9.5 Leader Transmitter

To transmit data to a follower, the leader must operate as a leader transmitter. Table 22.4. I2C Leader Transmitter shows the states the I²C module goes through while acting as a leader transmitter. Every state where an interaction is required has the possible interactions listed, along with the result of the interactions. The table also shows which interrupt flags are set in the different states. The interrupt flags enclosed in parenthesis may be set. If the BUSHOLD interrupt in I2C_IF is set, the module is waiting for an interaction, and the bus is frozen. The value of I2C_STATE will be equal to the values given in the table when the BUSHOLD interrupt flag is set, and can be used to determine which interaction is required to make the transmission continue.

The interrupt flag START in I2C_IF is set when the I²C module transmits the START.

A leader operation is started by issuing a START command by setting START in I2C_CMD. ADDR+W, i.e., the address of the follower + the R/W bit is then required by the I²C module. If this is not available in the transmit buffer, then the bus is held and the BUSHOLD interrupt flag is set. The value of I2C_STATE will then be 0x67. As seen in the table, the I²C module also stops in this state if the address is not available after a repeated start condition.

To continue, write a byte to I2C_TXDATA with the address of the follower in the 7 most significant bits and the least significant bit cleared (ADDR+W). This address will then be transmitted, and the follower will reply with an ACK or a NACK. If no follower replies to the address, the response will also be NACK. If the address was acknowledged, the leader now has four choices. It can send data by placing it in I2C_TXDATA (the leader should check the TXBL interrupt flag before writing to the transmit buffer), this data is then transmitted. The leader can also stop the transmission by sending a STOP, it can send a repeated start by sending START, or it can send a STOP and then a START as soon as possible. If the leader wishes to make another transfer immediately after the current, the preferred way is to start a new transfer directly by transmitting a repeated START instead of a STOP followed by a START. This is so because if a STOP is sent out, then any leader wishing to initiate a transfer on the bus can try to gain control of it.

If a NACK was received, the leader has to issue a CONT command in addition to providing data in order to continue transmission. This is not standard I²C, but is provided for flexibility. The rest of the options are similar to when an ACK was received.

If a new byte was transmitted, an ACK or NACK is received after the transmission of the byte, and the leader has the same options as for when the address was sent.

The leader may lose arbitration at any time during transmission. In this case, the ARBLOST interrupt flag in I2C_IF is set. If the arbitration was lost during the transfer of an address, and SLAVE in I2C_CTRL is set, the leader then checks which address was transmitted. If it was the address of the leader, then the leader goes to follower mode.

After a leader has transmitted a START and won any arbitration, it owns the bus until it transmits a STOP. After a STOP, the bus is released, and arbitration decides which bus leader gains the bus next. The MSTOP interrupt flag in I2C_IF is set when a STOP condition is transmitted by the leader.

Table 22.4. I2C Leader Transmitter

I2C_STATE	Description	I2C_IF	Required interaction	Response
0x67	Start transmitted	START interrupt flag (BUSHOLD interrupt flag)	ADDR+W -> TXDATA	ADDR+W will be sent
			STOP	STOP will be sent and bus released.
			STOP + START	STOP will be sent and bus released. Then a START will be sent when bus becomes idle.
0x67	Repeated start transmitted	START interrupt flag (BUSHOLD interrupt flag)	ADDR+W -> TXDATA	ADDR+W will be sent
			STOP	STOP will be sent and bus released.
			STOP + START	STOP will be sent and bus released. Then a START will be sent when bus becomes idle.
-	ADDR+W transmitted	TXFL interrupt flag (TXC interrupt flag)	None	

I2C_STATE	Description	I2C_IF	Required interaction	Response
0x97	ADDR+W transmitted, ACK received	ACK interrupt flag (BUSHOLD interrupt flag)	TXDATA	DATA will be sent
			STOP	STOP will be sent. Bus will be released
			START	Repeated start condition will be sent
			STOP + START	STOP will be sent and the bus released. Then a START will be sent when the bus becomes idle
0x9F	ADDR+W transmitted,NACK received	NACK (BUSHOLD interrupt flag)	CONT + TXDATA	DATA will be sent
			STOP	STOP will be sent. Bus will be released
			START	Repeated start condition will be sent
			STOP + START	STOP will be sent and the bus released. Then a START will be sent when the bus becomes idle
-	Data transmitted	TXFL interrupt flag (TXC interrupt flag)	None	
0xD7	Data transmitted,ACK received	ACK interrupt flag (BUSHOLD interrupt flag)	TXDATA	DATA will be sent
			STOP	STOP will be sent. Bus will be released
			START	Repeated start condition will be sent
			STOP + START	STOP will be sent and the bus released. Then a START will be sent when the bus becomes idle
0xDF	Data transmitted,NACK received	NACK(BUSHOLD interrupt flag)	CONT + TXDATA	DATA will be sent
			STOP	STOP will be sent. Bus will be released
			START	Repeated start condition will be sent
			STOP + START	STOP will be sent and the bus released. Then a START will be sent when the bus becomes idle
-	Stop transmitted	MSTOP interrupt flag	None	
			START	START will be sent when bus becomes idle
-	Arbitration lost	ARBLOST interrupt flag	None	
			START	START will be sent when bus becomes idle

22.3.9.6 Leader Receiver

To receive data from a follower, the leader must operate as a leader receiver, see Table 22.5. I2C Leader Receiver. This is done by transmitting ADDR+R as the address byte instead of ADDR+W, which is transmitted to become a leader transmitter. The address byte loaded into the data register thus has to contain the 7-bit follower address in the 7 most significant bits of the byte, and have the least significant bit set.

When the address has been transmitted, the leader receives an ACK or a NACK. If an ACK is received, the ACK interrupt flag in I2C_IF is set, and if space is available in the receive shift register, reception of a byte from the follower begins. If the receive buffer and shift register is full however, the bus is held until data is read from the receive buffer or another interaction is made. Note that the STOP and START interactions have a higher priority than the data-available interaction, so if a STOP or START command is pending, the highest priority interaction will be performed, and data will not be received from the follower.

If a NACK was received, the CONT command in I2C_CMD has to be issued in order to continue receiving data, even if there is space available in the receive buffer and/or shift register.

After a data byte has been received the leader must ACK or NACK the received byte. If an ACK is pending or AUTOACK in I2C_CTRL is set, an ACK is sent automatically and reception continues if space is available in the receive buffer.

If a NACK is sent, the CONT command must be used in order to continue transmission. If an ACK or NACK is issued along with a START or STOP or both, then the ACK/NACK is transmitted and the reception is ended. If START in I2C_CMD is set alone, a repeated start condition is transmitted after the ACK/NACK. If STOP in I2C_CMD is set, a stop condition is sent regardless of whether START is set. If START is set in this case, it is set as pending.

As when operating as a leader transmitter, arbitration can be lost as a leader receiver. When this happens the ARBLOST interrupt flag in I2C_IF is set, and the leader has a possibility of being selected as a follower given the correct conditions.

Table 22.5. I2C Leader Receiver

I2C_STATE	Description	I2C_IF	Required interaction	Response
0x63	START transmitted	START interrupt flag (BUSHOLD interrupt flag)	ADDR+R -> TXDATA	ADDR+R will be sent
			STOP	STOP will be sent and bus released.
			STOP + START	STOP will be sent and bus released. Then a START will be sent when bus becomes idle.
0x67	Repeated START transmitted	START interrupt flag(BUSHOLD interrupt flag)	ADDR+R -> TXDATA	ADDR+R will be sent
			STOP	STOP will be sent and bus released.
			STOP + START	STOP will be sent and bus released. Then a START will be sent when bus becomes idle.
-	ADDR+R transmitted	TXFL interrupt flag (TXC interrupt flag)	None	
0xA3	ADDR+R transmitted, ACK received	ACK interrupt flag(BUSHOLD)	RXDATA	Start receiving
			STOP	STOP will be sent and the bus released
			START	Repeated START will be sent
			STOP + START	STOP will be sent and the bus released. Then a START will be sent when the bus becomes idle
0x9B	ADDR+R transmitted,NACK received	NACK(BUSHOLD)	CONT + RXDATA	Continue, start receiving
			STOP	STOP will be sent and the bus released
			START	Repeated START will be sent
			STOP + START	STOP will be sent and the bus released. Then a START will be sent when the bus becomes idle

I2C_STATE	Description	I2C_IF	Required interaction	Response
0xB3	Data received	RXDATA interrupt flag(BUSHOLD interrupt flag)	ACK + RXDATA	ACK will be transmitted, reception continues
			NACK + CONT + RXDATA	NACK will be transmitted, reception continues
			ACK/NACK + STOP	ACK/NACK will be sent and the bus will be released.
			ACK/NACK + START	ACK/NACK will be sent, and then a repeated start condition.
			ACK/NACK + STOP + START	ACK/NACK will be sent and the bus will be released. Then a START will be sent when the bus becomes idle
-	Stop received	MSTOP interrupt flag	None	
			START	START will be sent when bus becomes idle
-	Arbitration lost	ARBLOST interrupt flag	None	
			START	START will be sent when bus becomes idle

22.3.9.7 SDA/SCL Status Monitor

The I^2C module supports an SDA and SCL monitoring function. Note that this functionality is only supported when the I2C module is in single leader mode, and when the follower doesn't use clock stretching. Additionally, firmware should set the ARBDIS bit in I2C_CTRL when using the SDA/SCL monitoring to prevent the bus being released.

The SDA monitor is enabled by setting the SDAMONEN in I2C_CTRL. Once enabled, the SDA monitor will check the status of the SDA line at the following points:

- At a Start Condition, before SDA falls
- At Stop Condition, after SDA rises

After checking, the monitor will set the SDAERR flag in I2C_IF it fails to read SDA==1. To allow the SDAERR flag to generate an IRQ, set the SDAERR bit in I2C_IEN.

Similarly, the SCL monitor is enabled by setting the SCLMONEN in I2C_CTRL. Once enabled, the SCL monitor will check the status of the SCL line at the following points:

- At a Start Condition, before SCL falls
- At every clock cycle, before SCL falls
- At Stop Condition, after SCL rises

After checking, the monitor will set the SCLERR flag in I2C_IF it fails to read SCL==1. To allow the SCLERR flag to generate an IRQ, set the SCLERR bit in I2C_IEN.

22.3.10 Bus States

The I2C_STATE register can be used to determine which state the I²C module and the I²C bus are in at a given time. The register consists of the STATE bit-field, which shows which state the I²C module is at in any ongoing transmission, and a set of single-bits, which reveal the transmission mode, whether the bus is busy or idle, and whether the bus is held by this I²C module waiting for a software response.

The possible values of the STATE field are summarized in Table 22.6. I2C STATE Values. When this field is cleared, the I²C module is not a part of any ongoing transmission. The remaining status bits in the I2C_STATE register are listed in Table 22.7. I2C Transmission Status.

Table 22.6. I2C STATE Values

Mode	Value	Description
IDLE	0	No transmission is being performed by this module.
WAIT	1	Waiting for idle. Will send a start condition as soon as the bus is idle.
START	2	Start transmit phase
ADDR	3	Address transmit or receive phase
ADDRACK	4	Address ACK/NACK transmit or receive phase
DATA	5	Data transmit or receive phase
DATAACK	6	Data ACK/NACK transmit or receive phase

Table 22.7. I2C Transmission Status

Bit	Description
BUSY	Set whenever there is activity on the bus. Whether or not this module is responsible for the activity cannot be determined by this byte.
MASTER	Set when operating as a leader. Cleared at all other times.
TRANSMITTER	Set when operating as a transmitter; either a leader transmitter or a follower transmitter. Cleared at all other times
BUSHOLD	Set when the bus is held by this I²C module because an action is required by software.
NACK	Only valid when bus is held and STATE is ADDRACK or DATAACK. In that case it is set if a NACK was received. In all other cases, the bit is cleared.

Note: I2C_STATE reflects the internal state of the I²C module, and therefore only held constant as long as the bus is held, i.e., as long as BUSHOLD in I2C_STATUS is set.

22.3.11 Follower Operation

The I²C module operates in leader mode by default. To enable follower operation, i.e., to allow the device to be addressed as an I²C follower, the SLAVE bit in I2C_CTRL must be set. In this case the I²C module operates in a mixed mode, both capable of starting transmissions as a leader, and being addressed as a follower. When operating in the follower mode, I2CCLK frequency must be higher than 2 MHz for Standard-mode, 5 MHz for Fast-mode, and 14 MHz for Fast-mode Plus.

22.3.11.1 Follower State Machine

The follower state machine is shown in Figure 22.16. I2C Follower State Machine. The dotted lines show where I²C-specific interrupt flags are set. The full-drawn circles show places where interaction may be required by software to let the transmission proceed.

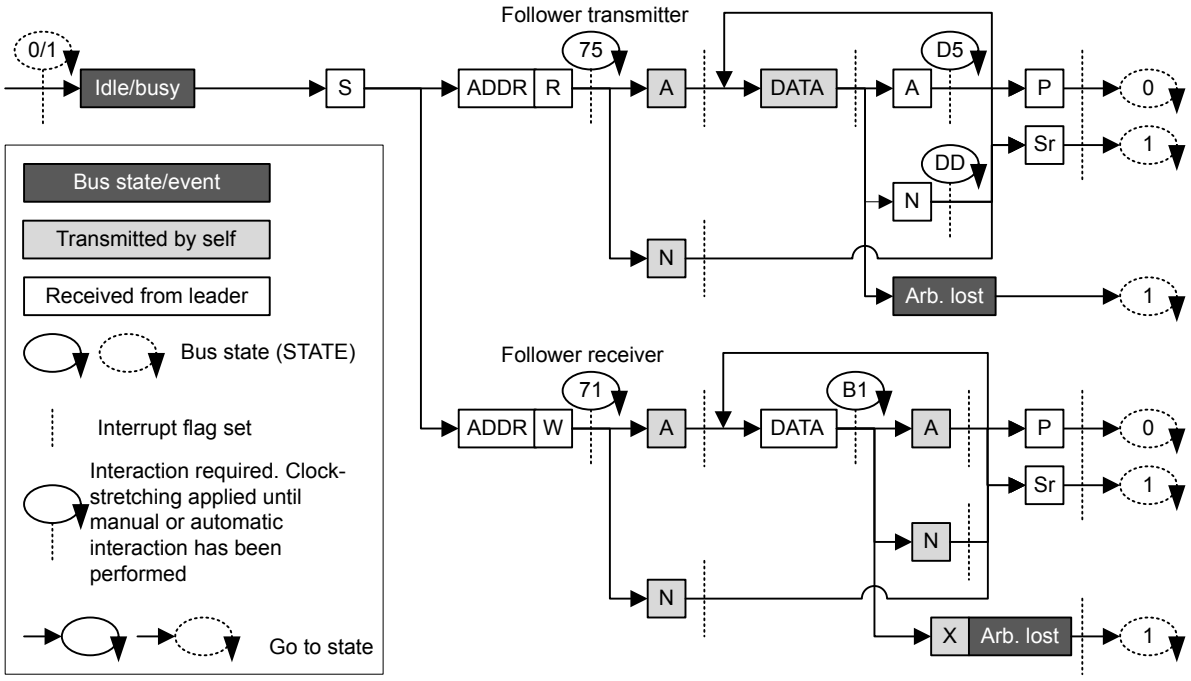


Figure 22.16. I2C Follower State Machine

22.3.11.2 Address Recognition

The I²C module provides automatic address recognition for 7-bit addresses. 10-bit address recognition is not fully automatic, but can be assisted by the 7-bit address comparator as shown in 22.3.13 Using 10-bit Addresses.

The follower address, i.e., the address which the I²C module should be addressed with, is defined in the I2C_SADDR register. In addition to the address, a mask must be specified, telling the address comparator which bits of an incoming address to compare with the address defined in I2C_SADDR. The mask is defined in I2C_SADDRMASK, and for every zero in the mask, the corresponding bit in the follower address is treated as a don't-care, i.e., the 0-masked bits are ignored.

An incoming address that fails address recognition is automatically replied to with a NACK. Since only the bits defined by the mask are checked, a mask with a value 0x00 will result in all addresses being accepted. A mask with a value 0x7F will only match the exact address defined in I2C_SADDR, while a mask 0x70 will match all addresses where the three most significant bits in I2C_SADDR and the incoming address are equal.

If GCAMEN in I2C_CTRL is not set, the start-byte, i.e., the general call address with the R/W bit set is ignored unless it is included in the defined follower address and the address mask.

When an address is accepted by the address comparator, the decision of whether to ACK or NACK the address is passed to software.

22.3.11.3 Follower Transmitter

When SLAVE in I2C_CTRL is set, the RSTART interrupt flag in I2C_IF will be set when repeated START conditions are detected. After a START or repeated START condition, the bus leader will transmit an address along with an R/W bit. If there is no room in the receive shift register for the address, the bus will be held by the follower until room is available in the shift register. Transmission then continues and the address is loaded into the shift register. If this address does not pass address recognition, it is automatically NACK'ed by the follower, and the follower goes to an idle state. The address byte is in this case discarded, making the shift register ready for a new address. It is not loaded into the receive buffer.

If the address was accepted and the R/W bit was set (R), indicating that the leader wishes to read from the follower, the follower now goes into the follower transmitter mode. Software interaction is now required to decide whether the follower wants to acknowledge the request or not. The accepted address byte is loaded into the receive buffer like a regular data byte. If no valid interaction is pending, the bus is held until the follower responds with a command. The follower can reject the request with a single NACK command.

The follower will in that case go to an idle state, and wait for the next start condition. To continue the transmission, the follower must make sure data is loaded into the transmit buffer and send an ACK. The loaded data will then be transmitted to the leader, and an ACK or NACK will be received from the leader.

Data transmission can also continue after a NACK if a CONT command is issued along with the NACK. This is not standard I^2C however.

If the leader responds with an ACK, it may expect another byte of data, and data should be made available in the transmit buffer. If data is not available, the bus is held until data is available.

If the response is a NACK however, this is an indication of that the leader has received enough bytes and wishes to end the transmission. The follower now automatically goes idle, unless CONT in I2C_CMD is set and data is available for transmission. The latter is not standard I^2C.

The leader ends the transmission by sending a STOP or a repeated START. The SSTOP interrupt flag in I2C_IF is set when the leader transmits a STOP condition. If the transmission is ended with a repeated START, then the SSTOP interrupt flag is not set.

Note: The SSTOP interrupt flag in I2C_IF will be set regardless of whether the follower is participating in the transmission or not, as long as SLAVE in I2C_CTRL is set and a STOP condition is detected

If arbitration is lost at any time during transmission, the ARBLOST interrupt flag in I2C_IF is set, the bus is released and the follower goes idle.

See Table 22.8. I2C Follower Transmitter for more information.

Table 22.8. I2C Follower Transmitter

I2C_STATE	Description	I2C_IF	Required interaction	Response
0x01	Repeated START received	RSTART interrupt flag (BUSHOLD interrupt flag)	RXDATA	Receive and compare address
0x75	ADDR + R received	ADDR interrupt flag	ACK + TXDATA	ACK will be sent, then DATA
		RXDATA interrupt flag	NACK	NACK will be sent, follower goes idle
		(BUSHOLD interrupt flag)	NACK + CONT + TXDATA	NACK will be sent, then DATA.
-	Data transmitted	TXBL interrupt flag (TXC interrupt flag)	None	
0xD5	Data transmitted, ACK received	ACK interrupt flag (BUSHOLD interrupt flag)	TXDATA	DATA will be transmitted
0xDD	Data transmitted, NACK received	NACK interrupt flag	None	The follower goes idle
		(BUSHOLD interrupt flag)	CONT + TXDATA	DATA will be transmitted

I2C_STATE	Description	I2C_IF	Required interaction	Response
-	Stop received	SSTOP interrupt flag	None	The follower goes idle
			START	START will be sent when bus becomes idle
-	Arbitration lost	ARBLOST interrupt flag	None	The follower goes idle
			START	START will be sent when the bus becomes idle

22.3.11.4 Follower Receiver

A follower receiver operation is started in the same way as a follower transmitter operation, with the exception that the address transmitted by the leader has the R/W bit cleared (W), indicating that the leader wishes to write to the follower. The follower then goes into follower receiver mode.

To receive data from the leader, the follower should respond to the address with an ACK and make sure space is available in the receive buffer. Transmission will then continue, and the follower will receive a byte from the leader.

If a NACK is sent without a CONT, the transmission is ended for the follower, and it goes idle. If the follower issues both the NACK and CONT commands and has space available in the receive buffer, it will be open for continuing reception from the leader.

When a byte has been received from the leader, the follower must ACK or NACK the byte. The responses here are the same as for the reception of the address byte.

The leader ends the transmission by sending a STOP or a repeated START. The SSTOP interrupt flag is set when the leader transmits a STOP condition. If the transmission is ended with a repeated START, then the SSTOP interrupt flag in I2C_IF is not set.

Note: The SSTOP interrupt flag in I2C_IF will be set regardless of whether the follower is participating in the transmission or not, as long as SLAVE in I2C_CTRL is set and a STOP condition is detected

If arbitration is lost at any time during transmission, the ARBLOST interrupt flag in I2C_IF is set, the bus is released and the follower goes idle.

See Table 22.9. I2C - Follower Receiver for more information.

Table 22.9. I2C - Follower Receiver

I2C_STATE	Description	I2C_IF	Required interaction	Response
0x01	Repeated START received	RSTART interrupt flag (BUSHOLD interrupt flag)	RXDATA	Receive and compare address
0x71	ADDR + W received	ADDR interrupt flag RXDATA interrupt flag (BUSHOLD interrupt flag)	ACK + RXDATA	ACK will be sent and data will be received
			NACK	NACK will be sent, follower goes idle
			NACK + CONT + RXDATA	NACK will be sent and DATA will be received.
0xB1	Data received	RXDATA interrupt flag (BUSHOLD interrupt flag)	ACK + RXDATA	ACK will be sent and data will be received
			NACK	NACK will be sent and follower will go idle
			NACK + CONT + RXDATA	NACK will be sent and data will be received
-	Stop received	SSTOP interrupt flag	None	The follower goes idle
			START	START will be sent when bus becomes idle
-	Arbitration lost	ARBLOST interrupt flag	None	The follower goes idle
			START	START will be sent when the bus becomes idle

22.3.12 Transfer Automation

The I^2C can be set up to complete transfers with a minimal amount of interaction.

22.3.12.1 DMA

DMA can be used to automatically load data into the transmit buffer and load data out from the receive buffer. When using DMA, software is thus relieved of moving data to and from memory after each transferred byte.

22.3.12.2 Automatic ACK

When AUTOACK in I2C_CTRL is set, an ACK is sent automatically whenever an ACK interaction is possible and no higher priority interactions are pending.

22.3.12.3 Automatic STOP

A STOP can be generated automatically on two conditions. These apply only to the leader transmitter.

If AUTOSN in I2C_CTRL is set, the I^2C module ends a transmission by transmitting a STOP condition when operating as a leader transmitter and a NACK is received.

If AUTOSE in I2C_CTRL is set, the I^2C module always ends a transmission when there is no more data in the transmit buffer. If data has been transmitted on the bus, the transmission is ended after the (N)ACK has been received by the follower. If a START is sent when no data is available in the transmit buffer and AUTOSE is set, then the STOP condition is sent immediately following the START. Software must thus make sure data is available in the transmit buffer before the START condition has been fully transmitted if data is to be transferred.

22.3.13 Using 10-bit Addresses

When using 10-bit addresses in follower mode, set the I2C_SADDR register to 1111 0XX where XX are the two most significant bits of the 10-bit address, and set I2C_SADDRMASK to 0xFF. Address matches will now be given on all 10-bit addresses where the two most significant bits are correct.

When receiving an address match, the follower must acknowledge the address and receive the first data byte. This byte contains the second part of the 10-bit address. If it matches the address of the follower, the follower should ACK the byte to continue the transmission, and if it does not match, the follower should NACK it.

When the leader is operating as a leader transmitter, the data bytes will follow after the second address byte. When the leader is operating as a leader receiver however, a repeated START condition is sent after the second address byte. The address sent after this repeated START is equal to the first of the address bytes transmitted previously, but now with the R/W byte set, and only the follower that found a match on the entire 10-bit address in the previous message should ACK this address. The repeated start should take the leader into a leader receiver mode, and after the single address byte sent this time around, the follower begins transmission to the leader.

22.3.14 Error Handling

Note: Some registers in the I^2C module are considered static. This means that these need to be set before an I^2C transaction starts and need to stay stable during the entire transaction.

Specifically:

- The GCAMEN and SLAVE fields in the I2C_CTRL register
- The I2C_SADDR register
- The GPIO_I2Cn_ROUTEEN, GPIO_I2Cn_SCLROUTE, and GPIO_I2Cn_SDAROUTE registers

22.3.14.1 ABORT Command

Some bus errors may require software intervention to be resolved. The I^2C module provides an ABORT command, which can be set in I2C_CMD, to help resolve bus errors.

When the bus for some reason is locked up and the I^2C module is in the middle of a transmission it cannot get out of, or for some other reason the I^2C wants to abort a transmission, the ABORT command can be used.

Setting the ABORT command will make the I^2C module discard any data currently being transmitted or received, release the SDA and SCL lines and go to an idle mode. ABORT effectively makes the I^2C module forget about any ongoing transfers.

22.3.14.2 Bus Reset

A bus reset can be performed by setting the START and STOP commands in I2C_CMD while the transmit buffer is empty. A START condition will then be transmitted, immediately followed by a STOP condition. A bus reset can also be performed by transmitting a START command with the transmit buffer empty and AUTOSE set.

22.3.14.3 I2C-Bus Errors

An I^2C-bus error occurs when a START or STOP condition is misplaced, which happens when the value on SDA changes while SCL is high during bit-transmission on the I^2C-bus. If the I^2C module is part of the current transmission when a bus error occurs, any data currently being transmitted or received is discarded, SDA and SCL are released, the BUSERR interrupt flag in I2C_IF is set to indicate the error, and the module automatically takes a course of action as defined in Table 22.10. I2C Bus Error Response.

Table 22.10. I2C Bus Error Response

	Misplaced START	Misplaced STOP
In a leader/follower operation	Treated as START. Receive address.	Go idle. Perform any pending actions.

22.3.14.4 Bus Lockup

A lockup occurs when a leader or follower on the I^2C-bus has locked the SDA or SCL at a low value, preventing other devices from putting high values on the bus, and thus making communication on the bus impossible.

Many follower-only devices operating on an I^2C-bus are not capable of driving SCL low, but in the rare case that SCL is stuck LOW, the advice is to apply a hardware reset signal to the followers on the bus. If this does not work, cycle the power to the devices in order to make them release SCL.

When SDA is stuck low and SCL is free, a leader should send 9 clock pulses on SCL while tristating the SDA. This procedure is performed in the GPIO module after clearing the GPIO_I2Cn_ROUTEEN register and disabling the I2C module. The device that held the bus low should release it sometime within those 9 clocks. If not, use the same approach as for when SCL is stuck, resetting and possibly cycling power to the followers.

Lockup of SDA can be detected by keeping count of the number of continuous arbitration losses during address transmission. If arbitration is also lost during the transmission of a general call address, i.e., during the transmission of the STOP condition, which should never happen during normal operation, this is a good indication of SDA lockup.

Detection of SCL lockups can be done using the timeout functionality defined in 22.3.14.6 Clock Low Timeout

22.3.14.5 Bus Idle Timeout

When SCL has been high for a significant amount of time, this is a good indication of that the bus is idle. On an SMBus system, the bus is only allowed to be in this state for a maximum of 50 µs before the bus is considered idle.

The bus idle timeout BITO in I2C_CTRL can be used to detect situations where the bus goes idle in the middle of a transmission. The timeout can be configured in BITO, and when the bus has been idle for the given amount of time, the BITO interrupt flag in I2C_IF is set. The bus can also be set idle automatically on a bus idle timeout. This is enabled by setting GIBITO in I2C_CTRL.

When the bus idle timer times out, it wraps around and continues counting as long as its condition is true. If the bus is not set idle using GIBITO or the ABORT command in I2C_CMD, this will result in periodic timeouts.

Note: This timeout will be generated even if SDA is held low.

The bus idle timeout is active as long as the bus is busy, i.e., BUSY in I2C_STATUS is set. The timeout can be used to get the I^2C module out of the busy-state it enters when reset, see 22.3.9.4 Reset State.

22.3.14.6 Clock Low Timeout

The clock timeout, which can be configured in CLTO in I2C_CTRL, starts counting whenever SCL goes low, and times out if SCL does not go high within the configured timeout. A clock low timeout results in CLTOIF in I2C_IF being set, allowing software to take action.

When the timer times out, it wraps around and continues counting as long as SCL is low. An SCL lockup will thus result in periodic clock low timeouts as long as SCL is low.

22.3.14.7 Clock Low Error

The I^2C module can continue transmission in parallel with another device for the entire transaction, as long as the two communications are identical. A case may arise when (before an arbitration has been decided upon) the I^2C module decides to send out a repeated START or a STOP condition while the other device is still sending data. In the I^2C protocol specifications, such a combination results in an undefined condition. The I^2C deals with this by generating a clock low error. This means that if the I^2C is transmitting a repeated START or a STOP condition and another device (another leader or a misbehaving follower) pulls SCL low before the I^2C sends out the START/STOP condition on SDA, a clock low error is generated. The CLERR interrupt flag is then set in the I2C_IF register, any held lines are released and the I^2C device goes to idle.

22.3.15 DMA Support

The I^2C module has full DMA support. A request for the DMA controller to write to the I^2C transmit buffer can come from TXFL (transmit buffer has room for more data). The DMA controller can write to the transmit buffer using the I2C_TXDATA register. DMA must be configured to transfer one byte of data when writing to I2C_TXDATA.

A request for the DMA controller to read from the I^2C receive buffer can come from RXDATAV (data available in the receive buffer). DMA must be configured to transfer one byte of data when reading from I2C_RXDATA.

22.3.16 Interrupts

The interrupts generated by the I^2C module are combined into one interrupt vector, I2Cn. If I^2C interrupts are enabled, an interrupt will be made if one or more of the interrupt flags in I2C_IF and their corresponding bits in I2C_IEN are set.

22.3.17 Wake-up

The I^2C receive section can be active in low energy modes down to EM1, and can wake up the CPU on address interrupt. All address match modes are supported.

22.4 I2C Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	I2C_IPVERSION	R	IP VERSION Register
0x004	I2C_EN	RW ENABLE	Enable Register
0x008	I2C_SWRST	RW SWRST	SWRST Register
0x00C	I2C_CTRL	RW LFSYNC	Control Register
0x010	I2C_CMD	W LFSYNC	Command Register
0x014	I2C_STATE	RH	State Register
0x018	I2C_STATUS	RH	Status Register
0x01C	I2C_CLKDIV	RW LFSYNC	Clock Division Register
0x020	I2C_SADDR	RW LFSYNC	Follower Address Register
0x024	I2C_SADDRMASK	RW LFSYNC	Follower Address Mask Register
0x028	I2C_RXDATA	RH	Receive FIFO Data Register
0x02C	I2C_RXDATAP	RH	Receive FIFO Data Peek Register
0x034	I2C_TXDATA	W	Transmit FIFO Data Register
0x03C	I2C_IF	RWH INTFLAG	Interrupt Flag Register
0x040	I2C_IEN	RW	Interrupt Enable Register
0x044	I2C_SYNCBUSY	RH	Syncbusy Status
0x1000	I2C_IPVERSION_SET	R	IP VERSION Register
0x1004	I2C_EN_SET	RW ENABLE	Enable Register
0x1008	I2C_SWRST_SET	RW SWRST	SWRST Register
0x100C	I2C_CTRL_SET	RW LFSYNC	Control Register
0x1010	I2C_CMD_SET	W LFSYNC	Command Register
0x1014	I2C_STATE_SET	RH	State Register
0x1018	I2C_STATUS_SET	RH	Status Register
0x101C	I2C_CLKDIV_SET	RW LFSYNC	Clock Division Register
0x1020	I2C_SADDR_SET	RW LFSYNC	Follower Address Register
0x1024	I2C_SADDRMASK_SET	RW LFSYNC	Follower Address Mask Register
0x1028	I2C_RXDATA_SET	RH	Receive FIFO Data Register
0x102C	I2C_RXDATAP_SET	RH	Receive FIFO Data Peek Register
0x1034	I2C_TXDATA_SET	W	Transmit FIFO Data Register
0x103C	I2C_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1040	I2C_IEN_SET	RW	Interrupt Enable Register
0x1044	I2C_SYNCBUSY_SET	RH	Syncbusy Status
0x2000	I2C_IPVERSION_CLR	R	IP VERSION Register
0x2004	I2C_EN_CLR	RW ENABLE	Enable Register
0x2008	I2C_SWRST_CLR	RW SWRST	SWRST Register
0x200C	I2C_CTRL_CLR	RW LFSYNC	Control Register
0x2010	I2C_CMD_CLR	W LFSYNC	Command Register
0x2014	I2C_STATE_CLR	RH	State Register
0x2018	I2C_STATUS_CLR	RH	Status Register

Offset	Name	Type	Description
0x201C	I2C_CLKDIV_CLR	RW LFSYNC	Clock Division Register
0x2020	I2C_SADDR_CLR	RW LFSYNC	Follower Address Register
0x2024	I2C_SADDRMASK_CLR	RW LFSYNC	Follower Address Mask Register
0x2028	I2C_RXDATA_CLR	RH	Receive FIFO Data Register
0x202C	I2C_RXDATAP_CLR	RH	Receive FIFO Data Peek Register
0x2034	I2C_TXDATA_CLR	W	Transmit FIFO Data Register
0x203C	I2C_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2040	I2C_IEN_CLR	RW	Interrupt Enable Register
0x2044	I2C_SYNCBUSY_CLR	RH	Syncbusy Status
0x3000	I2C_IPVERSION_TGL	R	IP VERSION Register
0x3004	I2C_EN_TGL	RW ENABLE	Enable Register
0x3008	I2C_SWRST_TGL	RW SWRST	SWRST Register
0x300C	I2C_CTRL_TGL	RW LFSYNC	Control Register
0x3010	I2C_CMD_TGL	W LFSYNC	Command Register
0x3014	I2C_STATE_TGL	RH	State Register
0x3018	I2C_STATUS_TGL	RH	Status Register
0x301C	I2C_CLKDIV_TGL	RW LFSYNC	Clock Division Register
0x3020	I2C_SADDR_TGL	RW LFSYNC	Follower Address Register
0x3024	I2C_SADDRMASK_TGL	RW LFSYNC	Follower Address Mask Register
0x3028	I2C_RXDATA_TGL	RH	Receive FIFO Data Register
0x302C	I2C_RXDATAP_TGL	RH	Receive FIFO Data Peek Register
0x3034	I2C_TXDATA_TGL	W	Transmit FIFO Data Register
0x303C	I2C_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3040	I2C_IEN_TGL	RW	Interrupt Enable Register
0x3044	I2C_SYNCBUSY_TGL	RH	Syncbusy Status

22.5 I2C Register Description

22.5.1 I2C_IPVERSION - IP VERSION Register

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

22.5.2 I2C_EN - Enable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	Disabling status when the module is disabled.			
0	EN	0x0	RW	**module enable**
	The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.			

22.5.3 I2C_SWRST - SWRST Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Reset busy status**
	When SWRST command is issued, resetting logic sets RESETTING status immediately, and later it is cleared when reset process finishes			
0	SWRST	0x0	W	**Software Reset bit**
	A software reset command field resets the module back to the initial condition, similar to a power on reset condition			

22.5.4 I2C_CTRL - Control Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0				0x0				0x0	0x0		0x0			0x0		0x0				0x0			0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access			RW				RW				RW	RW		RW			RW		RW				RW			RW	RW	RW	RW	RW	RW	RW
Name			RXFIL				TXFIL				SDAMONEN	SCLMONEN		CLTO			GIBITO		BITO				CLHR			GCAMEN	ARBDIS	AUTOSN	AUTOSE	AUTOACK	SLAVE	CORERST

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29:26	RXFIL	0x0	RW	**RX FIFO Interrupt level**
	Set RX FIFO Interrupt Level			
25:22	TXFIL	0x0	RW	**TX FIFO Interrupt level**
	Set TX FIFO Interrupt Level			
21	SDAMONEN	0x0	RW	**SDA Monitor Enable**
	Set to enable SDA monitor feature. This will enable SDA rise check at loopback path. This monitor can not be enabled in Multi-Leader application			

Value	Mode	Description
0	DISABLE	Disable SDA Monitor
1	ENABLE	Enable SDA Monitor

20	SCLMONEN	0x0	RW	**SCL Monitor Enable**
	Set to enable SCL monitor feature. This will enable SCL rise check at loopback path. This monitor can not be enabled in Multi-Leader application			

Value	Mode	Description
0	DISABLE	Disable SCL monitor
1	ENABLE	Enable SCL monitor

19	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
18:16	CLTO	0x0	RW	**Clock Low Timeout**
	Use to generate a timeout when CLK has been low for the given amount of time. Wraps around and continues counting when the timeout is reached. The timeout value can be calculated by timeout = PCC/(Fscl x (Nlow + Nhigh))			

Value	Mode	Description
0	OFF	Timeout disabled
1	I2C40PCC	Timeout after 40 prescaled clock cycles. In standard mode at 100 kHz, this results in a 50us timeout.
2	I2C80PCC	Timeout after 80 prescaled clock cycles. In standard mode at 100 kHz, this results in a 100us timeout.
3	I2C160PCC	Timeout after 160 prescaled clock cycles. In standard mode at 100 kHz, this results in a 200us timeout.
4	I2C320PCC	Timeout after 320 prescaled clock cycles. In standard mode at 100 kHz, this results in a 400us timeout.

Bit	Name	Reset	Access	Description
	5	I2C1024PCC		Timeout after 1024 prescaled clock cycles. In standard mode at 100 kHz, this results in a 1280us timeout.
15	GIBITO	0x0	RW	**Go Idle on Bus Idle Timeout**
	When set, the bus automatically goes idle on a bus idle timeout, allowing new transfers to be initiated.			
	Value	Mode		Description
	0	DISABLE		A bus idle timeout has no effect on the bus state.
	1	ENABLE		A bus idle timeout tells the I2C module that the bus is idle, allowing new transfers to be initiated.
14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	BITO	0x0	RW	**Bus Idle Timeout**
	Use to generate a timeout when SCL has been high for a given amount time between a START and STOP condition. When in a bus transaction, i.e. the BUSY flag is set, a timer is started whenever SCL goes high. When the timer reaches the value defined by BITO, it sets the BITO interrupt flag. The BITO interrupt flag will then be set periodically as long as SCL remains high. The bus idle timeout is active as long as BUSY is set. It is thus stopped automatically on a timeout if GIBITO is set. It is also stopped a STOP condition is detected and when the ABORT command is issued. The timeout is activated whenever the bus goes BUSY, i.e. a START condition is detected. The timeout value can be calculated by timeout = PCC/(Fscl x (Nlow + Nhigh))			
	Value	Mode		Description
	0	OFF		Timeout disabled
	1	I2C40PCC		Timeout after 40 prescaled clock cycles. In standard mode at 100 kHz, this results in a 50us timeout.
	2	I2C80PCC		Timeout after 80 prescaled clock cycles. In standard mode at 100 kHz, this results in a 100us timeout.
	3	I2C160PCC		Timeout after 160 prescaled clock cycles. In standard mode at 100 kHz, this results in a 200us timeout.
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	CLHR	0x0	RW	**Clock Low High Ratio**
	Determines the values of (and ratio between) the low and high parts of the clock signal generated on SCL as leader.			
	Value	Mode		Description
	0	STANDARD		Nlow=4 and Nhigh=4, and the Nlow:Nhigh ratio is 4:4
	1	ASYMMETRIC		Nlow=6 and Nhigh=3, and the Nlow:Nhigh ratio is 6:3
	2	FAST		Nlow=11 and Nhigh=6, and the Nlow:Nhigh ratio is 11:6
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	GCAMEN	0x0	RW	**General Call Address Match Enable**
	Set to enable address match on general call in addition to the programmed follower address.			
	Value	Mode		Description
	0	DISABLE		General call address will be NACK'ed if it is not included by the follower address and address mask.
	1	ENABLE		When a general call address is received, a software response is required
5	ARBDIS	0x0	RW	**Arbitration Disable**

Bit	Name	Reset	Access	Description
	A leader or follower will not release the bus upon losing arbitration.			

Value	Mode	Description
0	DISABLE	When a device loses arbitration, the ARBIF interrupt flag is set and the bus is released.
1	ENABLE	When a device loses arbitration, the ARBIF interrupt flag is set, but communication proceeds.

Bit	Name	Reset	Access	Description
4	AUTOSN	0x0	RW	**Automatic STOP on NACK**
	Write to 1 to make a leader transmitter send a STOP when a NACK is received from a follower.			

Value	Mode	Description
0	DISABLE	Stop is not automatically sent if a NACK is received from a follower.
1	ENABLE	The leader automatically sends a STOP if a NACK is received from a follower.

Bit	Name	Reset	Access	Description
3	AUTOSE	0x0	RW	**Automatic STOP when Empty**
	Write to 1 to make a leader transmitter send a STOP when no more data is available for transmission.			

Value	Mode	Description
0	DISABLE	A stop must be sent manually when no more data is to be transmitted.
1	ENABLE	The leader automatically sends a STOP when no more data is available for transmission.

Bit	Name	Reset	Access	Description
2	AUTOACK	0x0	RW	**Automatic Acknowledge**
	Set to enable automatic acknowledges.			

Value	Mode	Description
0	DISABLE	Software must give one ACK command for each ACK transmitted on the I2C bus.
1	ENABLE	Addresses that are not automatically NACK'ed, and all data is automatically acknowledged.

Bit	Name	Reset	Access	Description
1	SLAVE	0x0	RW	**Addressable as Follower**
	Set this bit to allow the device to be selected as an I2C follower.			

Value	Mode	Description
0	DISABLE	All addresses will be responded to with a NACK
1	ENABLE	Addresses matching the programmed follower address or the general call address (if enabled) require a response from software. Other addresses are automatically responded to with a NACK.

Bit	Name	Reset	Access	Description
0	CORERST	0x0	RW	**Soft Reset the internal state registers**
	Set to reset the I2C_STATE register, and return the I2C module to the IDLE state. Must clear this bit to resume normal operation condition			

Value	Mode	Description
0	DISABLE	No change to internal state registers
1	ENABLE	Reset the internal state registers

22.5.5 I2C_CMD - Command Register

Offset	Bit Position																														
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																								W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)	W(nB)
Name																								CLEARRX	CLEARPC	CLEARTX	ABORT	CONT	NACK	ACK	STOP	START

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	CLEARRX	0x0	W(nB)	**Clear RX FIFO**
	Set to cleart RX FIFO			
7	CLEARPC	0x0	W(nB)	**Clear Pending Commands**
	Set to clear pending commands.			
6	CLEARTX	0x0	W(nB)	**Clear TX**
	Set to cleart RX FIFO			
5	ABORT	0x0	W(nB)	**Abort transmission**
	Abort the current transmission making the bus go idle. When used in combination with STOP, a STOP condition is sent as soon as possible before aborting the transmission. The stop condition is subject to clock synchronization.			
4	CONT	0x0	W(nB)	**Continue transmission**
	Set to continue transmission after a NACK has been received.			
3	NACK	0x0	W(nB)	**Send NACK**
	Set to transmit a NACK the next time an acknowledge is required.			
2	ACK	0x0	W(nB)	**Send ACK**
	Set to transmit an ACK the next time an acknowledge is required.			
1	STOP	0x0	W(nB)	**Send stop condition**
	Set to send stop condition as soon as possible.			
0	START	0x0	W(nB)	**Send start condition**
	Set to send start condition as soon as possible. If a transmission is ongoing and not owned, the start condition will be sent as soon as the bus is idle. If the current transmission is owned by this module, a repeated start condition will be sent. Use in combination with a STOP command to automatically send a STOP, then a START when the bus becomes idle.			

22.5.6 I2C_STATE - State Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0			0x0	0x0	0x0	0x0	0x1
Access																									R			R	R	R	R	R
Name																									STATE			BUSHOLD	NACKED	TRANSMITTER	MASTER	BUSY

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:5	STATE	0x0	R	**Transmission State**

The state of any current transmission. Cleared if the I2C module is idle.

Value	Mode	Description
0	IDLE	No transmission is being performed.
1	WAIT	Waiting for idle. Will send a start condition as soon as the bus is idle.
2	START	Start transmit phase
3	ADDR	Address transmit or receive phase
4	ADDRACK	Address ack/nack transmit or receive phase
5	DATA	Data transmit or receive phase
6	DATAACK	Data ack/nack transmit or receive phase

4	BUSHOLD	0x0	R	**Bus Held**

Set if the bus is currently being held by this I2C module.

3	NACKED	0x0	R	**Nack Received**

Set if a NACK was received and STATE is ADDRACK or DATAACK.

2	TRANSMITTER	0x0	R	**Transmitter**

Set when operating as a leader transmitter or a follower transmitter. When cleared, the system may be operating as a leader receiver, a follower receiver or the current mode is not known.

1	MASTER	0x0	R	**Leader**

Set when operating as an I2C leader. When cleared, the system may be operating as an I2C follower.

0	BUSY	0x1	R	**Bus Busy**

Set when the bus is busy. Whether the I2C module is in control of the bus or not has no effect on the value of this bit. When the MCU comes out of reset, the state of the bus is not known, and thus BUSY is set. Use the ABORT command or a bus idle timeout to force the I2C module out of the BUSY state.

22.5.7 I2C_STATUS - Status Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset												0x0	0x0	0x0	0x0				0x0				0x0	0x0	0x1	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access												R	R	R	R				R				R	R	R	R	R	R	R	R	R	R
Name												CLEARRXBUSY	CLEARTXBUSY	RXFL	RXFCNT				TXFCNT				RXFULL	RXDATAV	TXFL	TXC	PABORT	PCONT	PNACK	PACK	PSTOP	PSTART

Bit	Name	Reset	Access	Description
31:21	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
20	CLEARRXBUSY	0x0	R	**Clear RX FIFO busy**
	Clear RX FIFO busy status			
19	CLEARTXBUSY	0x0	R	**Clear TX FIFO busy**
	Clear TX FIFO busy status			
18	RXFL	0x0	R	**RX FIFO level**
	Indicates the level of the recieve fifo. If RXFIL==0, set when the RX FIFO has at least one more frame in it. If RXFIL==1, set when the RX FIFO at least two more frames in it.			
17:14	RXFCNT	0x0	R	**RX FIFO Count**
	RX FIFO count indicates valid entries in RXFIFO			
13:10	TXFCNT	0x0	R	**TX FIFO Count**
	TX FIFO count indicates valid entries in TXFIFO			
9	RXFULL	0x0	R	**RX FIFO Full**
	Set when the receive fifo is full. Cleared when the receive fifo is no longer full. When this bit is set, there is still room for one more frame in the receive shift register.			
8	RXDATAV	0x0	R	**RX Data Valid**
	Set when data is available in the receive fifo. Cleared when the receive fifo is empty.			
7	TXFL	0x1	R	**TX FIFO Level**
	Indicates the level of the transmit fifo. If TXFIL==0, set when the TX FIFO has space for at least one more frame. If TXFIL==1, set when the TX FIFO has space for at least two more frames.			
6	TXC	0x0	R	**TX Complete**
	Set when a transmission has completed and no more data is available in the transmit FIFO and shift register. Cleared when a new transmission starts.			
5	PABORT	0x0	R	**Pending abort**
	An abort is pending and will be transmitted as soon as possible.			
4	PCONT	0x0	R	**Pending continue**
	A continue is pending and will be transmitted as soon as possible.			
3	PNACK	0x0	R	**Pending NACK**
	A not-acknowledge is pending and will be transmitted as soon as possible.			
2	PACK	0x0	R	**Pending ACK**
	An acknowledge is pending and will be transmitted as soon as possible.			

Bit	Name	Reset	Access	Description
1	PSTOP	0x0	R	**Pending STOP**
	A stop condition is pending and will be transmitted as soon as possible.			
0	PSTART	0x0	R	**Pending START**
	A start condition is pending and will be transmitted as soon as possible.			

22.5.8 I2C_CLKDIV - Clock Division Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0								
Access																								RW								
Name																								DIV								

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	DIV	0x0	RW	**Clock Divider**
	Specifies the clock divider for the I2C. Note that DIV must be 1 or higher when follower is enabled.			

22.5.9 I2C_SADDR - Follower Address Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									ADDR							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:1	ADDR	0x0	RW	**Follower address**
	Specifies the follower address of the device.			
0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

22.5.10 I2C_SADDRMASK - Follower Address Mask Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									SADDRMASK							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:1	SADDRMASK	0x0	RW	**Follower Address Mask** Specifies the significant bits of the follower address. Setting the mask to 0x00 will match all addresses, while setting it to 0x7F will only match the exact address specified by ADDR.
0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

22.5.11 I2C_RXDATA - Receive FIFO Data Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									RXDATA							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	RXDATA	0x0	R	**RX Data** Use this register to read from the receive fifo. Fifo is emptied on read access.

22.5.12 I2C_RXDATAP - Receive FIFO Data Peek Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									RXDATAP							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	RXDATAP	0x0	R	**RX Data Peek**
	Use this register to read from the receive fifo. Fifo is not emptied on read access.			

22.5.13 I2C_TXDATA - Transmit FIFO Data Register

Offset	Bit Position																													
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W							
Name																									TXDATA							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	TXDATA	0x0	W	**TX Data**
	Use this register to write a byte to the transmit fifo.			

22.5.14 I2C_IF - Interrupt Flag Register

Offset	Bit Position																															
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access											RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name											RXFLIF	SDAERR	SCLERR	CLERR	RXFULL	SSTOP	CLTO	BITO	RXUF	TXOF	BUSHOLD	BUSERR	ARBLOST	MSTOP	NACK	ACK	RXDATAV	TXBL	TXC	ADDR	RSTART	START

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	RXFLIF	0x0	RW	**RX FIFO level Interrupt Flag**
	If RXFIL==0, set when the RX FIFO has at least one more frame in it. If RXFIL==1, set when the RX FIFO at least two more frames in it.			
20	SDAERR	0x0	RW	**SDA Error Interrupt Flag**
	Set when the SDA at loopback path is not equal to SDA output			
19	SCLERR	0x0	RW	**SCL Error Interrupt Flag**
	Set when the SCL at loopback path is not equal to SCL output			
18	CLERR	0x0	RW	**Clock Low Error Interrupt Flag**
	Set when the clock is pulled low before a START or a STOP condition could be transmitted.			
17	RXFULL	0x0	RW	**Receive Fifo Full Interrupt Flag**
	Set when the receive fifo becomes full.			
16	SSTOP	0x0	RW	**Follower STOP condition Interrupt Flag**
	Set when a STOP condition has been received. Will be set regardless of the follower being involved in the transaction or not.			
15	CLTO	0x0	RW	**Clock Low Timeout Interrupt Flag**
	Set on each clock low timeout. The timeout value can be set in CLTO bit field in the I2Cn_CTRL register.			
14	BITO	0x0	RW	**Bus Idle Timeout Interrupt Flag**
	Set on each bus idle timeout. The timeout value can be set in the BITO bit field in the I2Cn_CTRL register.			
13	RXUF	0x0	RW	**Receive FIFO Underflow Interrupt Flag**
	Set when data is read from the receive fifo through the I2Cn_RXDATA register while the receive fifo is empty.			
12	TXOF	0x0	RW	**Transmit FIFO Overflow Interrupt Flag**
	Set when data is written to the transmit fifo while the transmit fifo is full.			
11	BUSHOLD	0x0	RW	**Bus Held Interrupt Flag**
	Set when the bus becomes held by the I2C module.			
10	BUSERR	0x0	RW	**Bus Error Interrupt Flag**
	Set when a bus error is detected. The bus error is resolved automatically, but the current transfer is aborted.			
9	ARBLOST	0x0	RW	**Arbitration Lost Interrupt Flag**
	Set when arbitration is lost.			
8	MSTOP	0x0	RW	**Leader STOP Condition Interrupt Flag**
	Set when a STOP condition has been successfully transmitted. If arbitration is lost during the transmission of the STOP condition, then the MSTOP interrupt flag is not set.			

Bit	Name	Reset	Access	Description
7	NACK	0x0	RW	**Not Acknowledge Received Interrupt Flag**
	Set when a NACK has been received.			
6	ACK	0x0	RW	**Acknowledge Received Interrupt Flag**
	Set when an ACK has been received.			
5	RXDATAV	0x0	RW	**Receive Data Valid Interrupt Flag**
	Set when received data is half full			
4	TXBL	0x0	RW	**Transmit FIFO Level Interrupt Flag**
	If TXFIL==0, set when the TX FIFO has space for at least one more frame. If TXFIL==1, set when the TX FIFO has space for at least two more frames.			
3	TXC	0x0	RW	**Transfer Completed Interrupt Flag**
	Set when the transmit shift register becomes empty and there is no more data in the transmit fifo.			
2	ADDR	0x0	RW	**Address Interrupt Flag**
	Set when incoming address is accepted, i.e. own address or general call address is received.			
1	RSTART	0x0	RW	**Repeated START condition Interrupt Flag**
	Set when a repeated start condition is detected.			
0	START	0x0	RW	**START condition Interrupt Flag**
	Set when a start condition is successfully transmitted.			

22.5.15 I2C_IEN - Interrupt Enable Register

Offset	Bit Position																															
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access											RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name											RXFLIEN	SDAERR	SCLERR	CLERR	RXFULL	SSTOP	CLTO	BITO	RXUF	TXOF	BUSHOLD	BUSERR	ARBLOST	MSTOP	NACK	ACK	RXDATAV	TXBL	TXC	ADDR	RSTART	START

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	RXFLIEN	0x0	RW	**RX FIFO level interrupt enable**
	Enable for RXFLIF interrupt flag			
20	SDAERR	0x0	RW	**SDA Error Interrupt Flag**
	Set when SDA at loopback path is not equal to SDA output			
19	SCLERR	0x0	RW	**SCL Error Interrupt Flag**
	Set when SCL at loopback path is not equal to SCL output			
18	CLERR	0x0	RW	**Clock Low Error Interrupt Flag**
	Set when the clock is pulled low before a START or a STOP condition could be transmitted.			
17	RXFULL	0x0	RW	**Receive Fifo Full Interrupt Flag**
	Set when the receive Fifo becomes full.			
16	SSTOP	0x0	RW	**Follower STOP condition Interrupt Flag**
	Set when a STOP condition has been received. Will be set regardless of the follower being involved in the transaction or not.			
15	CLTO	0x0	RW	**Clock Low Timeout Interrupt Flag**
	Set on each clock low timeout. The timeout value can be set in CLTO bit field in the I2Cn_CTRL register.			
14	BITO	0x0	RW	**Bus Idle Timeout Interrupt Flag**
	Set on each bus idle timeout. The timeout value can be set in the BITO bit field in the I2Cn_CTRL register.			
13	RXUF	0x0	RW	**Receive FIFO Underflow Interrupt Flag**
	Set when data is read from the receive fifo through the I2Cn_RXDATA register while the receive fifo is empty.			
12	TXOF	0x0	RW	**Transmit FIFO Overflow Interrupt Flag**
	Set when data is written to the transmit fifo while the transmit fifo is full.			
11	BUSHOLD	0x0	RW	**Bus Held Interrupt Flag**
	Set when the bus becomes held by the I2C module.			
10	BUSERR	0x0	RW	**Bus Error Interrupt Flag**
	Set when a bus error is detected. The bus error is resolved automatically, but the current transfer is aborted.			
9	ARBLOST	0x0	RW	**Arbitration Lost Interrupt Flag**
	Set when arbitration is lost.			
8	MSTOP	0x0	RW	**Leader STOP Condition Interrupt Flag**
	Set when a STOP condition has been successfully transmitted. If arbitration is lost during the transmission of the STOP condition, then the MSTOP interrupt flag is not set.			
7	NACK	0x0	RW	**Not Acknowledge Received Interrupt Flag**

Bit	Name	Reset	Access	Description
	Set when a NACK has been received.			
6	ACK	0x0	RW	**Acknowledge Received Interrupt Flag**
	Set when an ACK has been received.			
5	RXDATAV	0x0	RW	**Receive Data Valid Interrupt Flag**
	Set when data is available in the receive fifo. Cleared automatically when the receive fifo is read.			
4	TXBL	0x0	RW	**Transmit FIFO Level Interrupt Flag**
	Enable for TXFLIF interrupt flag			
3	TXC	0x0	RW	**Transfer Completed Interrupt Flag**
	Set when the transmit shift register becomes empty and there is no more data in the transmit fifo.			
2	ADDR	0x0	RW	**Address Interrupt Flag**
	Set when incoming address is accepted, i.e. own address or general call address is received.			
1	RSTART	0x0	RW	**Repeated START condition Interrupt Flag**
	Set when a repeated start condition is detected.			
0	START	0x0	RW	**START condition Interrupt Flag**
	Set when a start condition is successfully transmitted.			

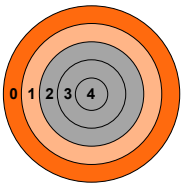
22.5.16 I2C_SYNCBUSY - Syncbusy Status

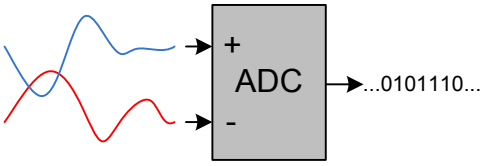
Offset	Bit Position																															
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset										0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access										R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Name										CLEARPC	ABORT	CONT	NACK	ACK	STOP	START	SDAMONEN	SCLMONEN	CLTO	GIBITO	BITO	CLHR	GCAMEN	ARBDIS	AUTOSN	AUTOSE	AUTOACK	SLAVE	CORERST	SADDRMASK	ADDR	DIV

Bit	Name	Reset	Access	Description
31:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	CLEARPC	0x0	R	**Sync busy of CLEARPC**
21	ABORT	0x0	R	**Sync busy of ABORT**
20	CONT	0x0	R	**Sync busy of CONT**
19	NACK	0x0	R	**Sync busy of NACK**
18	ACK	0x0	R	**Sync busy of ACK**
17	STOP	0x0	R	**Sync busy of STOP**
16	START	0x0	R	**Sync busy of START**
15	SDAMONEN	0x0	R	**Sync busy of SDAMONEN**
14	SCLMONEN	0x0	R	**Sync busy of SCLMONEN**
13	CLTO	0x0	R	**Sync busy of CLTO**
12	GIBITO	0x0	R	**Sync busy of GIBITO**
11	BITO	0x0	R	**Sync busy of BITO**
10	CLHR	0x0	R	**Sync busy of CLHR**
9	GCAMEN	0x0	R	**Sync busy of GCAMEN**
8	ARBDIS	0x0	R	**Sync busy of ARBDIS**
7	AUTOSN	0x0	R	**Sync busy of AUTOSN**
6	AUTOSE	0x0	R	**Sync busy of AUTOSE**
5	AUTOACK	0x0	R	**Sync busy of AUTOACK**
4	SLAVE	0x0	R	**Sync busy of SLAVE**
3	CORERST	0x0	R	**Sync busy of CORERST**
2	SADDRMASK	0x0	R	**Sync busy of SADDRMASK**
1	ADDR	0x0	R	**Sync busy of SADDR**

Bit	Name	Reset	Access	Description
0	DIV	0x0	R	**Syncbusy of CLKDIV**

23. ADC - Analog to Digital Converter





Quick Facts

What?

The ADC is used to convert analog voltages into a digital representation and features high-speed, low-power operation.

Why?

Many applications require measuring analog signals and recording them in a digital representation without exhausting the energy source.

How?

The ADC samples one or more input channels in a programmable sequence. With the help of PRS and DMA, the ADC can operate without CPU intervention. The ADC can be automatically shut down between conversions to reduce the energy consumption.

23.1 Introduction

The ADC uses a Succesive Approximation Register (SAR) architecture, with a resolution of 12 bits operating at up to one million samples per second (1 Msps). An integrated input multiplexer and channel sequencer can select from external I/Os and several internal signals.

23.2 Features

- 12 bit SAR architecture with adjustable acquisition time
- Up to 1 Msps sample rate
- Single ended or differential operation
- Data FIFO with 16 entries
- Internal and external conversion trigger sources
  - Immediate (software triggered)
  - Local ADC timer
  - External TIMER module (synchronous with output / PWM generation)
  - General PRS hardware signal
- Integrated prescaler for conversion clock generation
- Selectable reference sources
  - Internal reference
  - External precision reference
  - Analog supply
- Support for offset and gain calibration
- Programmable input gain: 5/16, 1/2, 1, 2, or 4x
- Hardware accumulation and averaging, 1 to 1024 samples
- Flexible output formatting
  - Programmable left or right justification
  - Optional channel ID tag
- Digital window comparison function detects when results are inside/outside a programmable window
- Two independent groups of configuration registers for setting acquisition time, analog gain, and hardware accumulator
- Autonomous multi-channel scan
  - Up to 16 configurable slots in scan sequence
  - Each slot allows independent selection of configuration group, channel selection, and window compare enable
  - One shot or continuous mode
  - Programmable watermark level to generate interrupts or initiate DMA transfers
- Available interrupt sources:
  - DVL (data valid level) FIFO watermark (also generates DMA request)
  - Window comparator conditions true
  - Single conversion completed
  - Scan table completed
  - FIFO overflow or underflow
  - Polarity Error interrupt
  - Port Allocation Error interrupt

23.3 Functional Description

The ADC module block diagram is shown in Figure 23.1. ADC Overview.

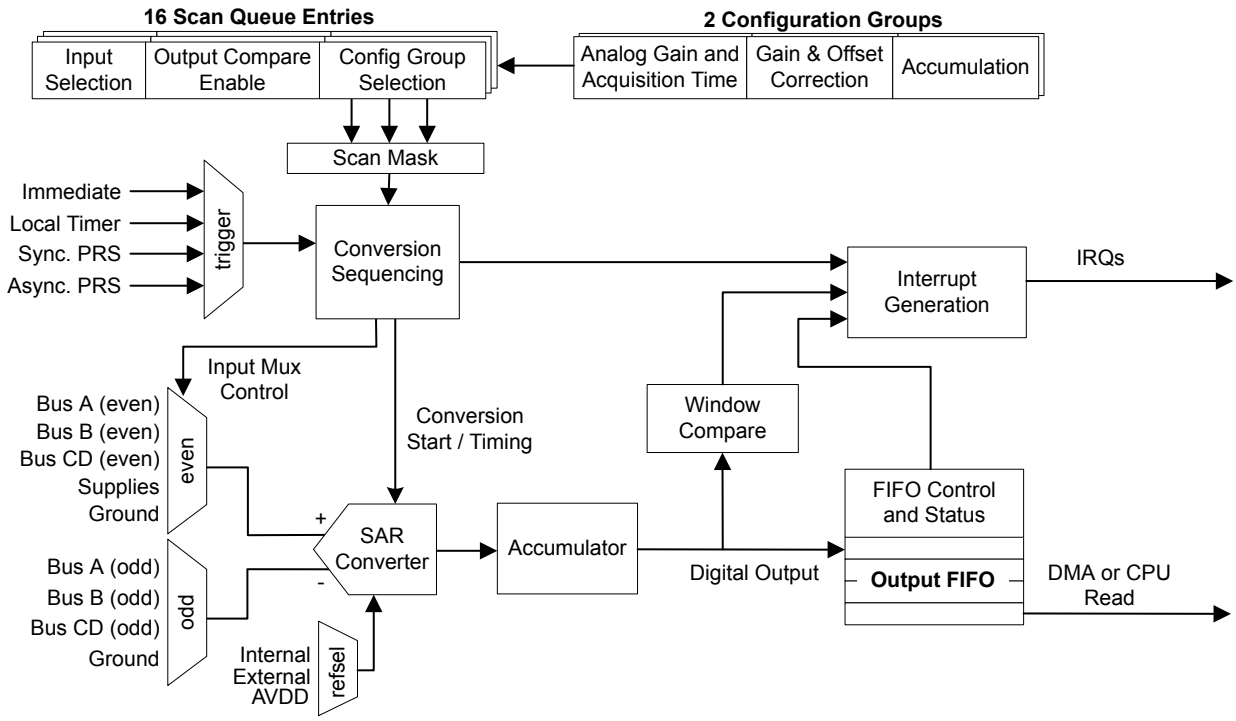


Figure 23.1. ADC Overview

23.3.1 Register Access

Many of the ADC module's configuration registers can only be written while the module is disabled (ADC_EN.EN = 0). These are ADC_CTRL, ADC_TIMER, ADC_CMPTHR, ADC_TRIGGER, ADC_CFGx, ADC_OFFSETSEx, ADC_OFFSETDIFFx, ADC_GAINCALx, ADC_SCANFIFOCFG, and ADC_SCANx.

23.3.2 Clocking

The ADC registers and FIFO read logic are clocked using the associated APB inferface clock (LSPCLK). The logic and operation of the ADC are clocked by CLK_SRC_ADC and CLK_CORE_ADC, both of which are derived from CLK_CMU_ADC, as shown in Figure 23.2. Clocking.

CLK_CMU_ADC is the incoming clock routed to the ADC by the CMU, and may be up to 40 MHz. It is selected within the CMU module. If the ADC is to be used synchronously with an external TIMER module, the clock must be configured to derive from EM01GRPACLK, which also clocks the TIMER modules.

CLK_SRC_ADC is derived from CLK_CMU_ADC, and must be no faster than 40 MHz. The HSCLKRATE field in ADC_CTRL sets the prescaler to divide CLK_CMU_ADC. The ADC's internal timer block as well as interrupt generation and other functions are clocked from CLK_CMU_ADC. If CLK_CMU_ADC is already 40 MHz or slower, HSCLKRATE can be set to 0x0 to pass the clock through to CLK_SRC_ADC without dividing it.

CLK_CORE_ADC is derived from CLK_SRC_ADC and is used to drive the ADC state machine logic, including the warmup timer. The ADCPRESCALE bit field in ADC_CTRL sets the prescaler to divide CLK_SRC_ADC.

The analog frontend clock (CLK_SAR) is directly derived from CLK_CORE_ADC and will be the same frequency. CLK_SAR is nominally 20 MHz, and must be between 17 and 22 MHz for proper ADC operation.

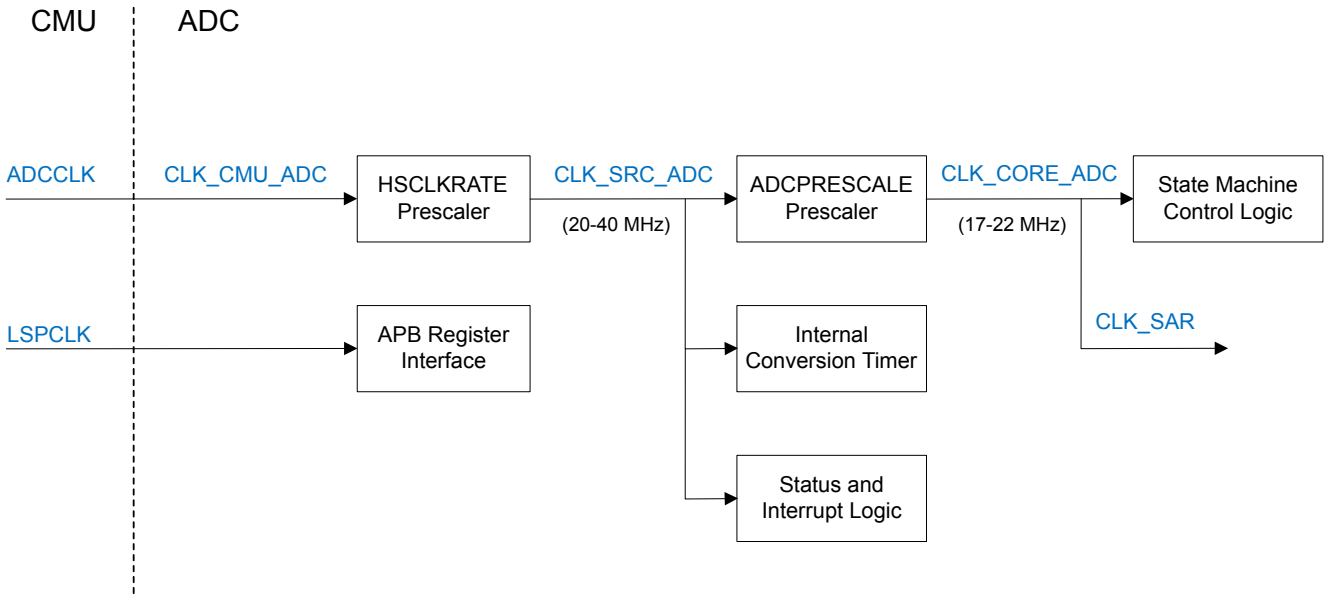


Figure 23.2. Clocking

23.3.3 Acquisition and Conversion Timing

Each ADC sample consists of two phases - acquisition and conversion. The timing of both phases is derived from CLK_SAR, which is directly derived from CLK_CORE_ADC.

When a conversion trigger is received and the ADC is already warmed up, the converter goes immediately to the acquisition phase. During acquisition the sampling capacitor is charged to the input voltage. The ADC requires a minimum acquisition time for proper settling, which is listed in the Device Data Sheet. The ATIME field in ADC_CFGx defines the acquisition time in CLK_CORE_ADC cycles (acquisition time = ATIME + 1 cycles). The ATIME field has a wide range to support sources with high series impedance. For high impedance sources the acquisition time should be adjusted to allow enough time for the internal sample capacitor to fully charge.

After acquisition, the sampling capacitor is disconnected from the input and the stored voltage is converted to a digital output word using successive approximation. The conversion time is dependent on CLK_SAR, which is directly derived from CLK_CORE_ADC. The conversion phase lasts for 13 CLK_SAR cycles.

If the channel sequencer does not need to change the channel at the end of a conversion cycle (for example during accumulated conversions, or during single-channel continuous conversions), the ADC will immediately begin the acquisition phase of the next sample.

If the channel sequencer is moving to the next slot at the end of the conversion phase, 4 cycles of CLK_CORE_ADC are used to change the multiplexer channel and configuration settings to prepare for the next sample.

Note that if the MASKREQ register is written during a continuous scan, this also requires a 4 cycle period between scans to update STMASK.

Note: The ADC should always be configured to adhere to the acquisition time (t_{ACQ}) and CLK_SAR frequency (f_{CLK_SAR}) limits specified in the data sheet. It is possible to configure the ADC to produce conversions faster than the maximum specified sample rate of 1 Msps. As long as f_{CLK_SAR} and t_{ACQ} limits are met, the ADC will perform as specified.

23.3.3.1 Warmup Time

To save energy, the ADC can be configured to power down completely or enter a standby state between scans, if full speed continuous operation is not required for the application. The minimum required ADC warm up time from a full powered-down state is 5 us. Warmup from a standby state requires at least 1 us. Warmup is automatically timed by the ADC logic when it is required, but software must configure the TIMEBASE field in ADC_CTRL for a minimum 1 us interval. The TIMEBASE counter receives CLK_CORE_ADC, and should be programmed accordingly. For example, if CLK_CORE_ADC is 20 MHz, TIMEBASE should be set to at least 0x13 (19) to produce the minimum 1 us interval. When transitioning from a powered-down state, the ADC will use five TIMEBASE intervals. When in standby the ADC will use one TIMEBASE interval.

The WARMUPMODE field in the ADC_CTRL register defines whether the ADC is powered down between scans (WARMUPMODE = NORMAL), in standby between scans (WARMUPMODE = KEEPINSTANDBY), or remains powered up (WARMUPMODE = KEEPWARM). The resulting start-up time is shown in Figure 23.3. Start-up Timing.

Note: On initial power-up, the ADC will use five TIMEBASE intervals of warmup time before the first conversion, even in WARMUPMODE = KEEPWARM or KEEPINSTANDBY. ADC_STATUS.ADCWARM reflects the current warmup status of the ADC.

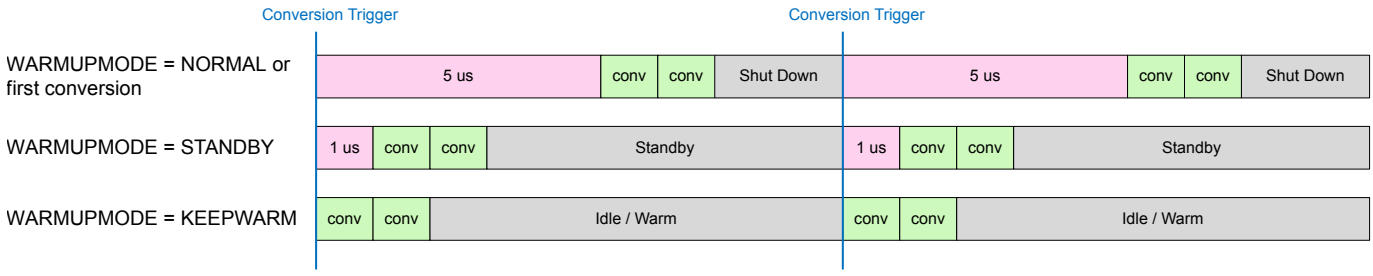


Figure 23.3. Start-up Timing

23.3.3.2 Scheduling and Triggers

The ADC has several triggering options available. Conversion triggering is configured using the SCANTRIGSEL bit field in the ADC_TRIGGER register. The options for trigger source are:

- IMMEDIATE - Trigger from software. This is useful for triggering conversions on-demand from software with no specific sampling frequency requirements, or initiating continuous conversions at full speed.
- TIMER - Use the ADC local timer to trigger conversions. This is useful for triggering conversions at precise intervals without using a general system TIMER resource.
- PRSCLKGRP - Use a synchronous PRS channel to trigger from an external peripheral in the same clock group domain (EM01GRPACLK). This is useful for synchronizing conversions precisely with external TIMER events or PWM outputs.

  Note: It is recommended to configure the PRS consumer registers prior to enabling synchronous PRS triggers to avoid false triggers.

- PRSPOS - Use a positive edge of an asynchronous PRS channel to trigger conversions. The trigger source will require 1-2 CLK_SRC_ADC cycles to synchronize. This is useful for triggering conversions as needed from asynchronous peripheral sources such as GPIO inputs, LETIMER events, etc.
- PRSNEG - Use a negative edge of an asynchronous PRS channel to trigger conversions. This is the same as PRSPOS, but operates on negative edges of the selected input.

The scan trigger source can be configured to generate one request per trigger, or begin continuous conversions. Setting SCANTRIGACTION to ONCE will make one request each time the selected trigger occurs, and the ADC will perform all conversions specified in the scan once, and then return to an idle or off state. The next conversion trigger will begin the sequence again.

Setting SCANTRIGACTION to CONTINUOUS instructs the conversion trigger to initiate continuous scans. When a scan cycle completes, a new one will be requested automatically, without requiring a new trigger. The ADC hardware also supports a limited programmable delay between scan cycles. If REPDELAY is set to NODELAY, the next scan will begin immediately without any delay. REPDELAY may also be set to a non-zero value, which will delay the start of the next scan by $2^{REPDELAY}$ CLK_CORE_ADC cycles, up to 1024 CLK_CORE_ADC cycles (at REPDELAY = 10).

The selected conversion trigger source will not be recognized until the ADC_CMD.SCANSTART command is sent to the converter. Note that for IMMEDIATE triggering, the SCANSTART command itself is the conversion trigger.

Once the SCANSTART command has been sent, any trigger detection will set the pending status bit ADC_STATUS.SCANQUEUEPENDING which indicates that the trigger has been recognized, but conversions have not yet started.

Conversions may be halted by sending the ADC_CMD.SCANSTOP command. If SCANSTOP is issued before conversions have begun (while SCANQUEUEPENDING is set), the conversion will be canceled. If the scanner has already started conversions when SCANSTOP is issued, the ADC will complete the current conversion before halting. Any remaining channels in the scan sequence not be converted after the SCANSTOP command, and a new SCANSTART will re-start the sequencer from the first channel.

23.3.3.2.1 Conversion Timing Examples

This section presents several examples to help calculate ADC timing in different situations. The examples are assuming that the converter is configured for continuous scanning to show the relationship between multiple conversions. These examples also assume the following:

- CLK_CORE_ADC and CLK_SAR are running at the nominal frequency of 20 MHz
- CFGx.ATIME is set to 6, producing 350 ns acquisition time

Continuous Single-Channel

In Figure 23.4. Continuous Single-Channel Conversion Example, the converter is configured with a single channel enabled in the MASKREQ register and no accumulation. The REPDELAY field is set to NODELAY. After the conversion trigger is detected, the ADC begins performing conversions, with 7 clocks of acquisition time and 13 clocks for the SAR conversion. The total time for each sample is 20 clocks and the converter is sampling at 1 MHz.

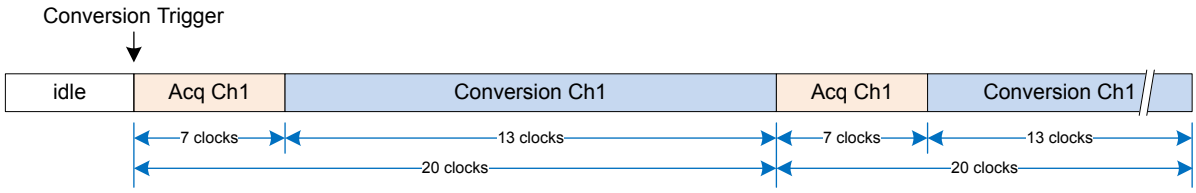


Figure 23.4. Continuous Single-Channel Conversion Example

Continuous Two-Channel

In Figure 23.4. Continuous Single-Channel Conversion Example, the converter is configured with two channels enabled in the MASKREQ register and no accumulation. The REPDELAY field is set to NODELAY. After the conversion trigger is detected, the ADC performs the conversion for the first channel, followed by the conversion for the second channel. Each acquisition + conversion lasts 20 clocks as in the previous example, and there is a 4-clock gap between conversions to change channels. In this example, the converter is producing output words at approximately 833 kHz, and the per-channel sample rate is approximately 416.6 kHz.

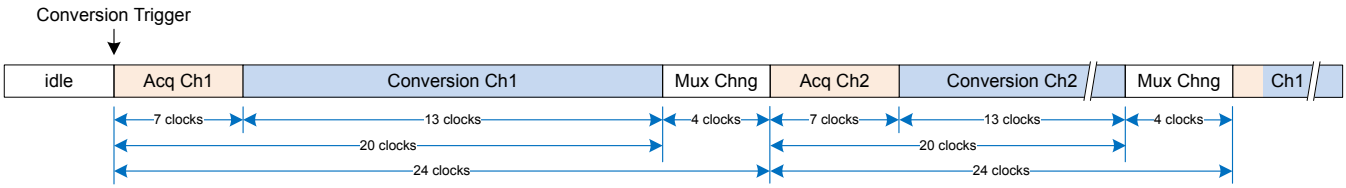


Figure 23.5. Continuous Two-Channel Conversion Example

Continuous Two-Channel with Accumulation

In Figure 23.4. Continuous Single-Channel Conversion Example, the converter is configured with two channels enabled in the MASKREQ register and 2-sample accumulation for each channel (CFGx.AVERAGESEL = X2). The REPDELAY field is set to NODELAY. After the conversion trigger is detected, the ADC performs two conversions on the first channel, followed by two conversions for the second channel. The samples are accumulated and the two samples on the first channel happen back to back at 20 clock cycles each, for a total of 40 clocks. There is a 4-clock gap to change channels, and the second channel is sampled in the same manner. In this example, the converter is producing output words at 454 kHz, and the per-channel sample rate is 227 kHz.

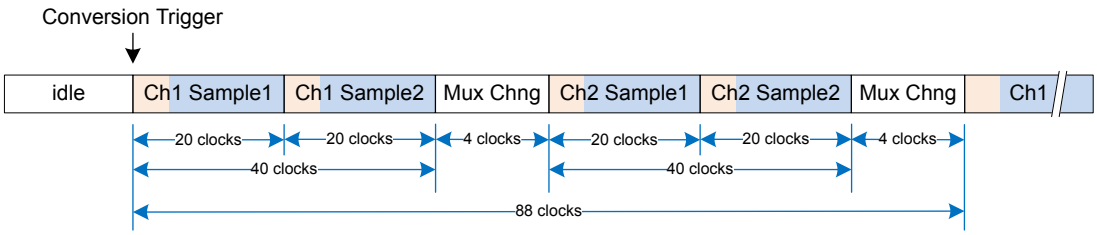


Figure 23.6. Continuous Two-Channel Conversions with Accumulation Example

Using the Repeat Delay

The REPDELAY field enables a programmable delay between scan cycles. In Figure 23.7. Continuous Two-Channel Conversions with Repeat Delay Example, the converter is configured with two channels enabled in the MASKREQ register and no accumulation. The REPDELAY field is set to a value other than NODELAY in order to place a delay between scan cycles. After the conversion trigger is detected, the ADC performs the conversion for the first channel, followed by the conversion for the second channel. There is a 4-clock gap between conversions to change channels. When the conversion phase for the final channel in the scan (channel 2) is complete, the repeat delay counter is activated. The delay will last for $2^{REPDELAY}$ CLK_CORE_ADC cycles, and a new scan will be triggered at the end of the delay.

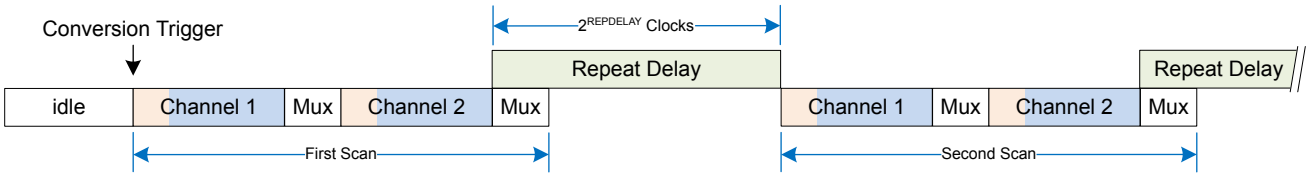


Figure 23.7. Continuous Two-Channel Conversions with Repeat Delay Example

Note: The repeat delay happens in parallel to a multiplexer switch event. When the multiplexer is switched at the end of the conversion cycle as shown in this example, REPDELAY must be set to a value of 2 or greater to ensure that the multiplexer switch completes before the beginning of the next acquisition cycle.

23.3.4 Reference Selection and Analog Gain

The default ADC reference is to use the internal voltage reference. The VREFP pin or analog power supply voltage can also be used as a voltage reference. The reference voltage is selected using the REFSEL field in ADC_CTRL.

Table 23.1. REFSEL

REFSEL (Reference selection)	Description	Voltage Range
VREFINT	Internal voltage reference	1.2 V
VREFPL	External Reference Pin (Low Voltage)	1.05 <= VREFP <= 1.25 V
VREFPH	External Reference Pin (High Voltage)	1.25 < VREFP <= AVDD
VDDA	Analog Power Supply	AVDD
VREFPBUF	External Reference Pin (Buffered)	1.15 <= VREFP <= 1.25

The ADC also has analog gain selection, controlled via the ANALOGGAIN field in ADC_CFGx. The analog gain can be set to 0.3125x, 0.5x, 1x, 2x, or 4x. The analog gain impacts where the full-scale input reading occurs. For example, with a 1.25 V external reference and ANALOGGAIN set to 2x, the analog input to the ADC is multiplied by a factor of 2, and a full-scale reading occurs at 1.25 V / 2 = 0.625 V. If ANALOGGAIN is set to 0.5x, the full-scale reading of the ADC will not occur until the input reaches 2.5 V.

Note that the actual ADC input voltage range may be limited to a lower voltage than the effective full scale. The ADC frontend is supplied by AVDD, while the GPIO pads are supplied by IOVDD. Any external inputs to the ADC (VREFP, or inputs routed to GPIO) will be limited by the lower of the two AVDD and IOVDD supplies. Internal signals connected to the ADC are limited only by the AVDD supply.

The sampling capacitance (C_S) is changed according to the analog gain setting.

Table 23.2. Input Sampling Capacitance vs. Analog Gain

Analog Gain Setting	Input Sampling Capacitance
0.3125x	0.375 pF
0.5x	0.6 pF
1x	1.2 pF
2x	2.4 pF
4x	4.8 pF

Given the sampling capacitance and the sample rate of the converter (f_{SAMPLE}), the effective input impedance of the converter can be calculated.

For differential mode, the impedance seen from AIN+ to AIN-:

$R_{IN(DIFF)} = 1 / (C_S * f_{SAMPLE})$

For single-ended mode, the impedance seen from AIN+ to VSS:

$R_{IN(SE)} = 1 / (2 * C_S * f_{SAMPLE})$

23.3.5 Input and Configuration Selection

The ADC supports measurement on a number of internal and external signals. External signals are routed to GPIO through shared ABUS resources on the device. Appropriate ABUS resources must be allocated to the ADC in the GPIO module before they can be selected.

The ADC_SCANx registers are used to select the inputs and configurations for each of the scan table's 16 entries, while the ADC_MASKREQ register defines which of the scan table slots will be converted during a scan.

In each ADC_SCANx register, the PORTPOS and PINPOS fields are used to select a signal for the positive ADC input, while PORTNEG and PINNEG are used to select a signal for the negative ADC input. The ADC_SCANx.CFG field selects which of the two configuration sets will be used with the input (i.e. configuration options specified in ADC_CFGx, ADC_OFFSETSEx, ADC_OFFSETDIFFx, and ADC_GAINCALx). ADC_SCANx.CMP defines whether the channel will be compared against the window comparator settings.

To perform single-ended conversions, the PORTNEG field should be set to GND. This indicates that the positive ADC input will be measured with reference to chip ground. PORTPOS and PINPOS should be used to select the desired input signal. The PINNEG field is not used for single-ended conversions.

Note: The negative reference for single-ended conversions is the VSS paddle on the device. This is the same pin that is used for the supply current return path for most of the device supplies. Impedance in the PCB ground path should be minimized to keep large supply current changes from impacting the ADC performance. A Kelvin connection between the signal source and the VSS paddle may also be used for this purpose.

To perform differential conversions, PORTPOS, PINPOS are used to select the positive input to the ADC, while PORTNEG and PINNEG are used to select the negative input. Note that the physical implementation of ABUS connections to the ADC input multiplexers limits some options. When performing differential conversions, the PINPOS and PINNEG fields must be from one even-numbered pin and one odd-numbered pin. Selecting two even pins or two odd pins for a differential measurement is not allowed. See 23.3.5.3 ABUS Input Selection Examples for more details.

At the beginning of a scan ADC_MASKREQ is copied to the ADC_STMASK register, which is used during that scan to select the converted channels. Although the SCANx registers cannot be changed on-the-fly, the ADC_MASKREQ register may be written during a scan to specify a different set of channels to be converted for the next scan. Writing to MASKREQ will set the ADC_STATUS.MASKREQWRITEPENDING bit to '1', indicating that MASKREQ holds new information that is not yet taken by STMASK. MASKREQWRITEPENDING should be checked to ensure it is '0' before writing a new value to MASKREQ.

23.3.5.1 External GPIO Inputs

GPIO input selections are routed through shared ABUS resources. In order for the ADC to use any GPIO as an input, the ADC must be allocated appropriate analog bus resources in the GPIO_ABUSALLOC, GPIO_BBUSALLOC, or GPIO_CDBUSALLOC registers. For example, if ADC0 will be using both odd and even numbered pins on GPIO port PA, then AEVEN0 and AODD0 in GPIO_ABUSALLOC could both be set to ADC0. This gives ADC0 access to these two buses. Generally, bus access is set to specific peripherals at configuration time and left alone - it is not normally required to change the bus allocation on the fly. If the ADC requests a pin from a bus that has not been allocated to the ADC, an error will be generated, the PORTALLOCERRIF in ADC_IF will be set, and any conversion result will be 0. For more details on analog bus structure and capabilities, refer to the GPIO section.

When the appropriate analog buses have been configured to route to the ADC, GPIO selection is a simple matter of programming the desired port and pin into the PORTPOS, PINPOS, PORTNEG, and PINNEG fields. For example, to configure a channel to convert the differential voltage between pins PA4 and PA5, PORTPOS = PORTA, PINPOS = 4, PORTNEG = PORTA, PINNEG = 5. If an invalid selection is made, a polarity error will be generated. More specific examples are described in 23.3.5.3 ABUS Input Selection Examples.

23.3.5.2 Internal and Dedicated Inputs

Internal signals and dedicated inputs are not routed through the shared ABUS resources. Internal resources are selected directly by the settings of PORTPOS and PORTNEG. When PORTPOS is set to SUPPLY, PINPOS is used to select which of the power supplies is connected. To facilitate power supply measurements of higher supply voltages using internal reference options, the supply voltages are attenuated by a factor of 4.

Table 23.3. Supply Selection (PORTPOS = SUPPLY)

PINPOS	Supply Connection	Voltage at Positive Input
0	AVDD	AVDD / 4
1	IOVDD	IOVDD / 4
2	DVDD	DVDD / 4
3	DECOUPLE	DECOUPLE / 4

23.3.5.3 ABUS Input Selection Examples

When configuring to measure a single-ended signal through the ABUS, the positive input selection should always point to the desired input, and PORTNEG should be programmed to GND.

Correct configuration examples for single-ended conversions are shown in Figure 23.8. Single-Ended Port/Pin Selection Even Channel and Figure 23.9. Single-Ended Port/Pin Selection Odd Channel.

Note that when a pin on an ODD ABUS is selected, the ADC logic will perform the conversion with the positive mux connected to GND and the negative mux connected to the selected GPIO, and the output is inverted.

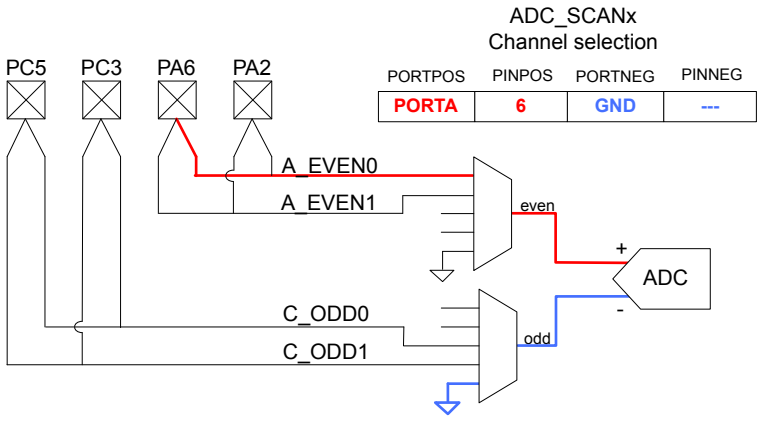


Figure 23.8. Single-Ended Port/Pin Selection Even Channel

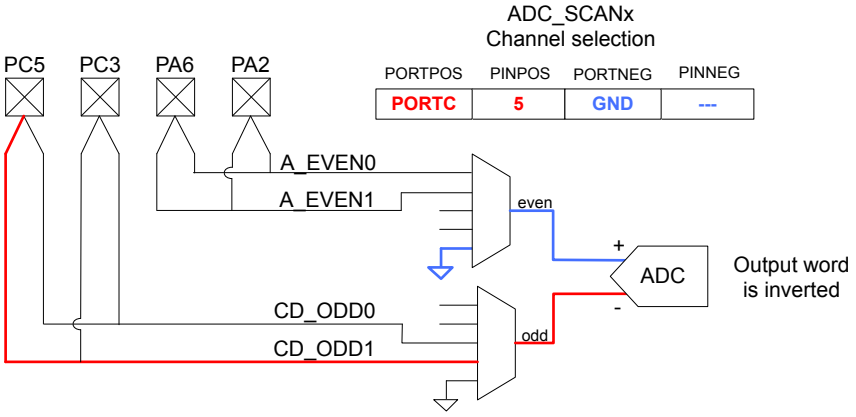


Figure 23.9. Single-Ended Port/Pin Selection Odd Channel

Figure 23.10. Single-Ended Port/Pin Selection Polarity Error shows an example where the PORTPOS input has been configured to GND, with PORTNEG and PINNEG configured for a GPIO pin. This will result in a polarity error (POLARITYERRIF in ADC_IF will be set) and any conversion result will be invalid.

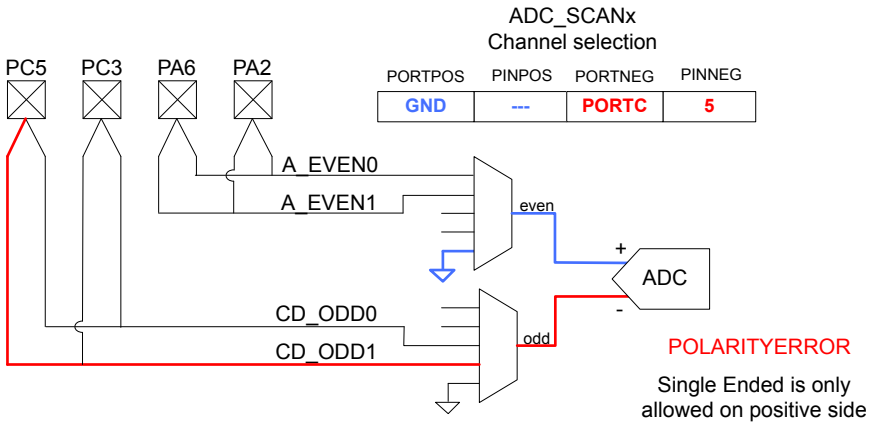


Figure 23.10. Single-Ended Port/Pin Selection Polarity Error

Correct configuration examples for differential conversions are shown in Figure 23.11. Differential Port/Pin Selection Even-Positive, Odd-Negative and Figure 23.12. Differential Port/Pin Selection Odd-Positive, Even-Negative. In both these examples, the inputs were selected from one EVEN ABUS and one ODD ABUS. As with single-ended mode, if an ODD ABUS is selected for the positive channel, the ADC logic will automatically invert the result to present it with the desired polarity.

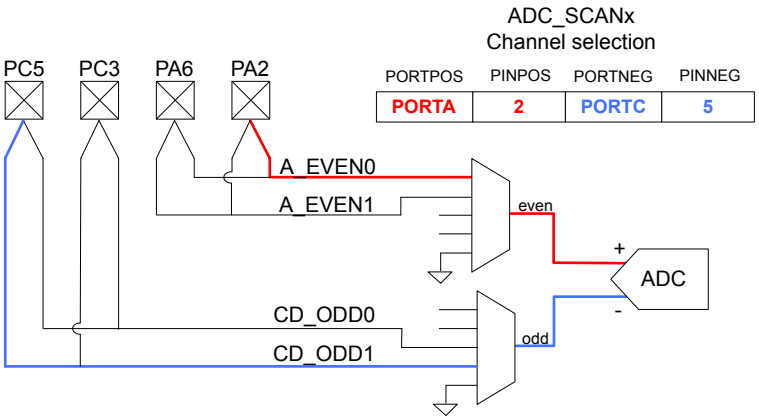


Figure 23.11. Differential Port/Pin Selection Even-Positive, Odd-Negative

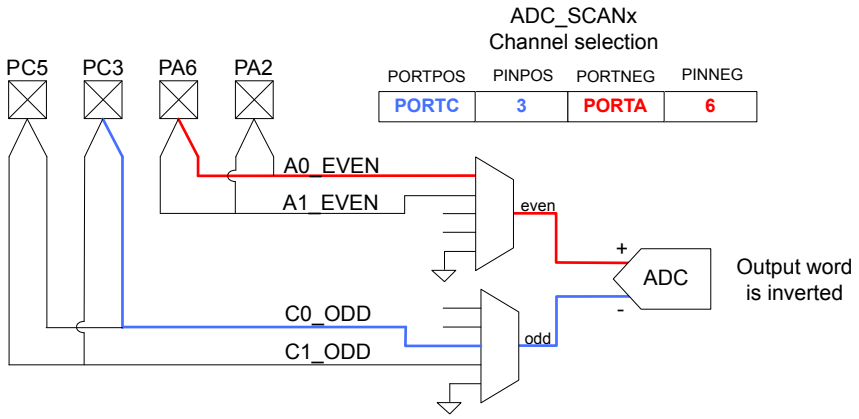


Figure 23.12. Differential Port/Pin Selection Odd-Positive, Even-Negative

Figure 23.13. Differential Port/Pin Selection Polarity Error shows an example where the both the positive and the negative input selections point to an ODD ABUS. Even if the ADC has been allocated both buses, they both route through the negative input multiplexer and cannot be measured against one another. This will result in a polarity error (POLARITYERRIF in ADC_IF will be set). Likewise, a polarity error will occur if both inputs are selected from an EVEN ABUS.

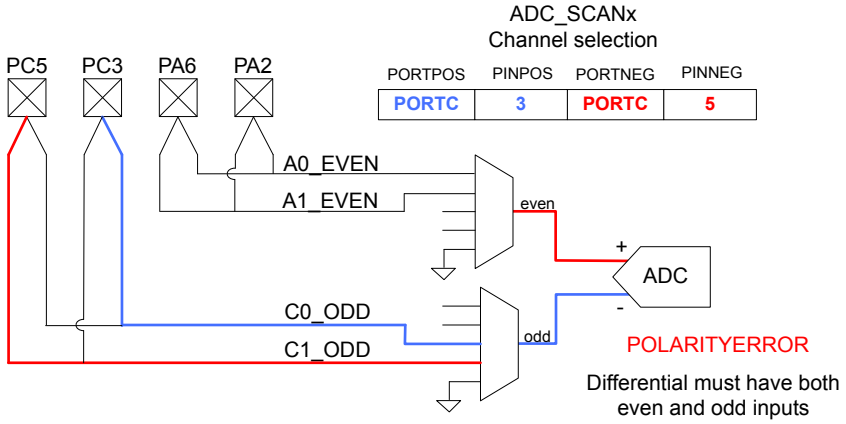


Figure 23.13. Differential Port/Pin Selection Polarity Error

23.3.5.4 Channel Sequencer

The channel sequencer (also called the scan queue) allows the ADC to convert up to 16 channels in sequence without CPU intervention. Input and configuration selection for each channel in the scan table is specified by the ADC_SCANx register for that channel (channel 0 is configured with ADC_SCAN0, channel 1 is configured with ADC_SCAN1, and so on). The ADC_MASKREQ register allows software to define which of the scan table entries to convert during a scan. Each channel is enabled by writing a '1' to the corresponding bit in ADC_MASKREQ (bit 0 enables ADC_SCAN0, bit 1 enables ADC_SCAN1, etc). For example, channels 0, 1, and 7 can be enabled by writing bits 0, 1, and 7 of ADC_MASKREQ to '1' (ADC_MASKREQ = 0x0083). Any channels enabled by MASKREQ are converted in numerical order from low to high during a scan, and the number of channels converted per scan is determined by the number of '1's in MASKREQ.

The ADC_SCANx registers must be configured when the ADC module is disabled (ADC_EN.EN = 0). ADC_MASKREQ can be written while ADC_EN.EN is set to '1'. If a scan operation is in progress, MASKREQ will be synchronized and held until the current scan operation has completed. Then MASKREQ is copied into the STMASK register for the next scan operation. ADC_STMASK is the working copy of the MASKREQ used by the ADC during a scan. MASKREQ will only transfer to STMASK when a scan is not in progress. ADC_STATUS.MASKREQWRITEPENDING can be used by software to see when the MASKREQ write has been transferred to STMASK. Writing a new MASKREQ in the middle of a scan will not corrupt the current scan. Software which writes to MASKREQ during a scan operation must ensure MASKREQWRITEPENDING returns to 0 before updating ADC_MASKREQ again.

23.3.6 Gain and Offset Correction

The ADC has built in gain and offset correction capabilities. Each of the two configuration groups contains its own correction values stored in the ADC_OFFSETSEx, ADC_OFFSETDIFFx, and ADC_GAINCALx registers, allowing the ADC to automatically apply the appropriate correction for the ADC configuration that is being used.

The calibration registers contain fields for adjusting the offset and gain for three different types of multiplexer connections:

1. **SEPOS** (Single-Ended Positive): These parameters are applied when a single-ended conversion is performed on an even-numbered GPIO pin or an internal signal such as a power supply. Such conversions are performed with the ADC's positive input connected to the signal of interest and the negative input connected to GND.
2. **SENEG** (Single-Ended Negative): These parameters are applied when a single-ended conversion is performed on an odd-numbered GPIO pin. Such conversions are performed with the ADC's negative input connected to the signal of interest and the positive input connected to GND. The results are inverted to produce a positive output word.
3. **DIFF** (Differential): These parameters are applied when a differential conversion is performed.

The calibration registers for a group should be configured with correction values that are appropriate for the selected ANALOGGAIN setting in that group.

The ADC will use the appropriate correction values from these fields based on the SCANx register channel selection.

23.3.6.1 Using Production Calibration Parameters

ADC calibration is performed on every device during Silicon Labs production test. Production calibration values are stored in the Device Info space in the ADCnCALDATA and ADCnOFFSET locations, where 'n' is the ADC instance. The stored calibration values must be adjusted according to the ANALOGGAIN setting to be used, and written into the appropriate correction registers.

Offset Correction Parameters

Offset correction parameters are stored in the ADCnOFFSET location in the Device Info space, in the TRIMOFF1X, TRIMOFF2X and TRIMOFF4X fields. The stored values are each a 6-bit number representing between -32 and +31 codes of offset trim. A value of 32 represents zero offset correction. The TRIMOFF1X value is to be used with ANALOGGAIN settings of 1x and lower. TRIMOFF2X and TRIMOFF4X are to be used when ANALOGGAIN is set to 2x or 4x, respectively.

The offset correction fields ADC_OFFSETSEx.SEPOS, ADC_OFFSETSEx.SENEG, and ADC_OFFSETDIFFx.DIFF are all 12-bit numbers with a range of -2048 to 2047. A value of 2048 represents zero offset correction. The 6-bit stored value can be translated to the 12-bit hardware field by adding 2016. The offset correction field values for each gain setting are summarized in Table 23.4. Offset Correction Fields.

Table 23.4. Offset Correction Fields

Analog Gain Setting	Value for offset fields: OFFSETSEx.SEPOS OFFSETSEx.SENEG	Value for offset field: OFFSETDIFFx.DIFF
0.3125x	TRIMOFF1X + 2016	TRIMOFF1X + 2017
0.5x	TRIMOFF1X + 2016	TRIMOFF1X + 2017
1x	TRIMOFF1X + 2016	TRIMOFF1X + 2017
2x	TRIMOFF2X + 2016	TRIMOFF2X + 2017
4x	TRIMOFF4X + 2016	TRIMOFF4X + 2017

Gain Correction Parameters

The gain correction fields ADC_GAINCALx.SEPOS, ADC_GAINCALx.SENEG, and ADC_GAINCALx.DIFF are encoded as 7-bit numbers. The five LSBs (bit 4-0) binary encoded value, and the two MSBs (bits 6-5) are thermometer-encoded with equal weight. The value 1100000 (binary) for gain settings 0.5x to 4x represents no gain correction, or a factor of 1.0. For the gain setting 0.3125x, 1.0 gain correction factor is represented by a value of 0100000 (binary). The gain correction step size is equal to 1/(1024*gain). For example, at gain = 1x, the gain can be adjusted in approximately 0.1% steps.

Gain correction parameters are stored in the ADCnCALDATA location in the Device Info space, in the TRIMGAIN4X and TRIMGAIN0X3INT fields. Each field is 7 bits wide, encoded with the same format as the gain correction fields.

When using a gain of 4x or 0.3125x, the gain trim values from Device Info can be directly used in the gain correction registers. Gain settings of 2x, 1x, and 0.5x use the TRIMGAIN4X value with modification. The effective gain step of the GAIN correction registers is indirectly proportional to the gain setting, and so the TRIMGAIN4X value must be adjusted to produce the same overall gain correction factor.

23.3.7 Output Data FIFOs

The ADC has a sixteen-word data FIFO for conversion results.

Conversion results are read from the FIFO using the ADC_SCANFIFODATA register. Reading SCANFIFODATA will pop the oldest result from the FIFO. It is also possible to read the most recent valid data word using ADC_SCANDATA. Reading SCANDATA does not pop a conversion from the FIFO.

When the scan FIFO has valid data SCANFIFODV in ADC_STATUS is set to 1. The data valid status bit is cleared automatically whenever the FIFO is empty. For more granular FIFO status, the number of data words present in the FIFO is indicated in ADC_SCANFIFOSTAT.

A programmable data level watermark is also available for the FIFO, allowing hardware to trigger interrupts or DMA operations when a specified number of conversion results are available. The DVL field in register SCANFIFOCFG sets the watermark level. If the number of valid entries in the FIFO reaches or exceeds the level set in DVL, the SCANFIFODVL flag in the ADC_IF register will be set to 1. If enabled, an interrupt or DMA request will be triggered when this flag is set.

Overflow and underflow status flags are also available in ADC_IF. An overflow condition occurs when an ADC conversion completes, but the FIFO is already full. In an overflow case the SCANFIFOOF flag in ADC_IF will be set. The most recent conversion will still be available in the SCANDATA register, but the FIFO will not be updated with the new data. An underflow condition occurs when software or hardware attempts to read from an empty FIFO. In an underflow case the SCANFIFOUF flag in ADC_IF will be set.

23.3.7.1 Data Alignment and Channel ID

The ADC has data alignment options and the ability to include a channel ID along with the conversion data. Alignment and channel ID are configured in the ADC_SCANFIFOCFG register.

The ALIGNMENT bit field specifies the data justification and the number of data bits as shown in Figure 23.14. Data Alignment.

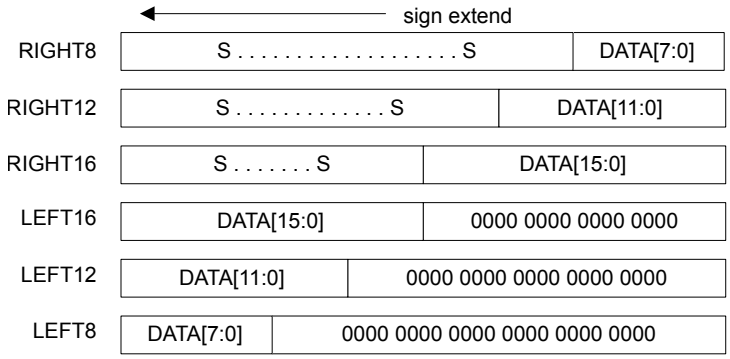


Figure 23.14. Data Alignment

The SHOWID bit controls whether the conversion channel ID is included in the output data word. This option is primarily used when converting multiple channels in a scan to help software determine which channel each conversion result came from. Figure 23.15. Data Alignment With ID shows output data formatting including the ID, when SHOWID = 1.

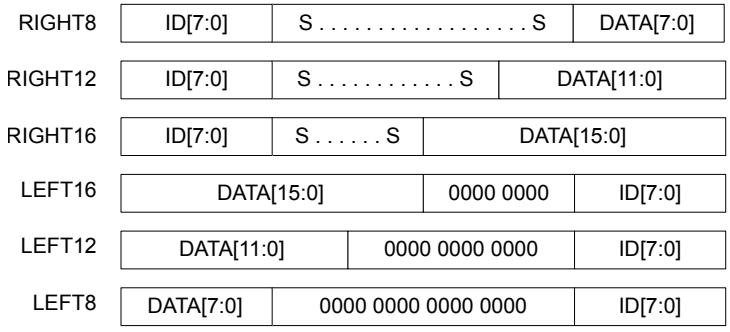


Figure 23.15. Data Alignment With ID

23.3.7.2 Output Polarity

The output polarity of the ADC depends on the type of conversion being performed. Single-ended conversions will produce unipolar output, and differential conversions will produce bipolar output.

Unipolar samples are unsigned integers representing zero to positive full-scale. Bipolar samples are two's-complement signed integers, representing negative full-scale to positive full-scale.

23.3.7.3 Digital Accumulation and Averaging

The ADC may optionally accumulate multiple conversion results before posting an output word to the FIFO. Digital accumulation is controlled by the AVERAGESEL field in the ADC_CFGx register. The ADC will accumulate $2^{AVERAGESEL}$ samples of the selected channel back-to-back. AVERAGESEL has a range between 0 and 10, meaning between 1 and 1024 conversions in powers of 2 can be accumulated per output. The ALIGNMENT field defines the placement of the DATA in the 32-bit output, and the number of bits available in the output word.

For ALIGNMENT = LEFT16 the MSB of the accumulated value will always be aligned in the MSB of the DATA field. For example, if a single conversion is performed (AVERAGESEL = 0), bits 15-4 of the DATA field will hold bits 11-0 of the accumulator, and bits 3-0 of DATA will be 0000. If AVERAGESEL is increased to 2 (four conversions), the accumulator now has four samples per output word, resulting two extra bits of information. In this case, bits 15-2 of the DATA field will hold bits 13-0 of the accumulator, and bits 1-0 of DATA will be 00.

On revision B devices and later, for ALIGNMENT = RIGHT16 the output word in the DATA field grows from the LSB upward, until the accumulated value is greater than 16 bits (when AVERAGESEL > 5). For example, if a single conversion is performed (AVERAGESEL = 0), bits 11-0 of the DATA field will hold bits 11-0 of the accumulator, and bits 15-12 of DATA will be sign-extended. If AVERAGESEL is increased to 2 (four conversions), the accumulator now has four samples per output word, resulting two extra bits of information. In this case, bits 13-0 of the DATA field will hold bits 13-0 of the accumulator, and bits 15-14 of DATA will be sign-extended.

Note: On revision A devices, the RIGHT16 option produces MSB-aligned results in the DATA field, as described for the the LEFT16 setting.

If the accumulator holds more bits than will fit into the selected DATA field, the LSBs will be truncated. For example, in the LEFT16 or RIGHT16 case, when AVERAGESEL is 5, the accumulator holds 17 bits of data after taking all 32 samples. Bits 15-0 of DATA will hold bits 16-1 of the accumulated result. In the case of LEFT8 or RIGHT8, DATA always holds the 8 MSBs of the accumulator, and in the case of LEFT12 or RIGHT12, DATA always holds the 12 MSBs of the accumulator.

23.3.7.4 Flushing the FIFOs

The data FIFO has a command bit in the ADC_CMD register that can be used to trigger a FIFO flush. To flush the FIFO:

1. The ADC must be enabled with the clock running.
2. Disable the conversion queue using the SCANSTOP bit in the ADC_CMD register.
3. Ensure the queue is disabled by reading the SCANQEN flag in the ADC_STATUS register.
4. Set the SCANFIFOFLUSH bit in the ADC_CMD register to flush the FIFO.
5. Wait for the SCANFIFOFLUSHING status bit in ADC_STATUS to go low.

23.3.8 Window Compare

The ADC has a window comparison unit that can trigger interrupts conditional on the output data of the converter. The window comparison unit has two thresholds - greater than or equal (ADGT), and less than or equal (ADLT), which are programmable through the ADC_CMPTHR register. The ADGT and ADLT thresholds always use a 16 bit, left-justified format, regardless of the format specified by the FIFO. A 12-bit aligned conversion result will be compared against the upper 12 bits of the window comparator.

The window comparison unit is active on the ADC output on a conversion-by-conversion basis. It is not possible to set different window comparison thresholds for different channels or groups. However, each channel specified in the ADC has a CMP bit field to enable the window comparison on results from that channel. For example, it is possible to only apply the window comparison and associated interrupt to scan channel #3 by setting the CMP field in ADC_SCAN3 to 1. When the CMP field associated with a channel is 0, the window comparator will not be active for results from that channel.

The window comparator supports conditional triggering on output results which are inside or outside a specified window. When ADLT is greater than or equal to ADGT, the comparator will trigger on an "inside" condition, or when DATA <= ADLT **and** DATA >= GT. When ADLT is less than ADGT, the comparator will trigger on an "outside" condition, or when DATA <= ADLT **or** DATA >= GT.

Figure 23.16. Window Comparison Examples shows different configurations of the ADLT and ADGT values and the resulting windows. When the window comparator detects that the appropriate conditions are met (shown by the shaded region in the figure), it will generate an interrupt via the SCANCMP flag in ADC_IF.

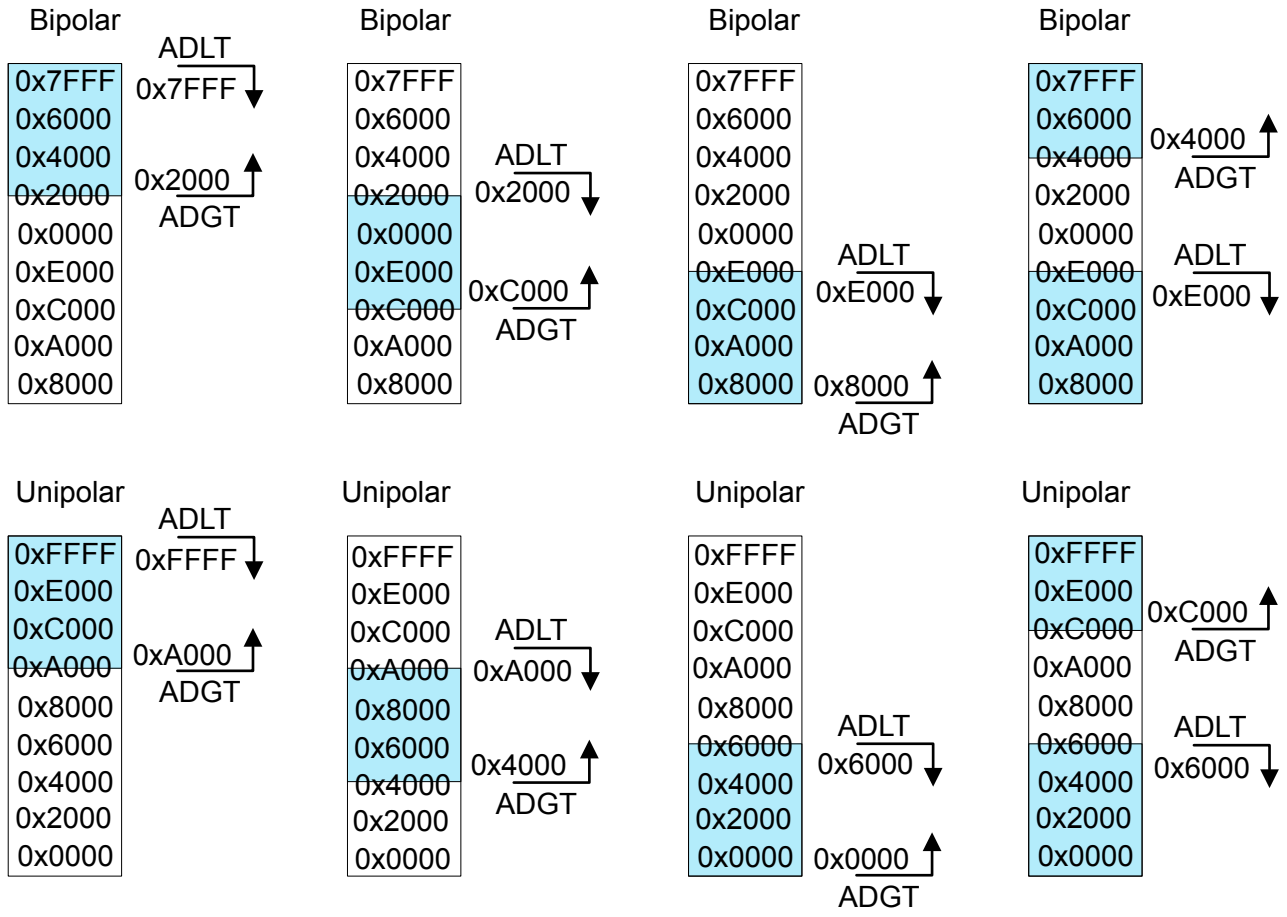


Figure 23.16. Window Comparison Examples

23.3.9 Interrupts

Interrupts are enabled in the ADC_IEN register, allowing interrupts to be generated on several different ADC conditions. Each of the flags in ADC_IF has a corresponding enable bit in the ADC_IEN register. A brief overview of the available interrupt sources is shown in the list below; more details can be found in the relevant sections of this chapter.

- SCANFIFODVL - The scan FIFO watermark specified in SCANFIFOCFG.DVL has been reached or exceeded.
- SCANCMP - A conversion result from the scan queue tripped the window comparator.
- SCANENTRYDONE - A scan queue conversion has completed.
- SCANTABLEDONE - A scan queue operation has completed (all channels specified in the scan mask have been converted once).
- POLARITYERR - A channel polarity selection error has occurred (two channels from the EVEN multiplexer or two channels from the ODD multiplexer were selected for positive and negative inputs).
- PORTALLOCERR - A port allocation error has occurred (a pin not allocated to the ADC in the GPIO bus allocation registers was requested).
- SCANFIFOOF - A scan FIFO overflow has occurred.
- SCANFIFOUF - A scan FIFO underflow has occurred.

Hardware sets the interrupt flags in ADC_IF, and the flags remain set (sticky) until cleared by software. The interrupt flags should be cleared before enabling the ADC to remove any previous interrupt history. Clearing or setting interrupt bits can be done by writing to ADC_IF with a set or clear mask.

23.4 ADC Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	ADC_IPVERSION	R	IPVERSION
0x004	ADC_EN	RW ENABLE	Enable
0x008	ADC_CTRL	RW CONFIG	Control
0x00C	ADC_CMD	W SYNC	Command
0x010	ADC_TIMER	RW CONFIG	Timer Control
0x014	ADC_STATUS	RH	Status
0x018	ADC_MASKREQ	RW SYNC	Select Channels to Be Converted
0x01C	ADC_STMASK	RH SYNC	Scan Table Mask
0x020	ADC_CMPTHR	RW CONFIG	Digital Window Comparator Threshold
0x024	ADC_IF	RWH INTFLAG	Interrupt Flags
0x028	ADC_IEN	RW	Interrupt Enable
0x02C	ADC_TRIGGER	RW CONFIG	Trigger
0x050	ADC_CFGx	RW CONFIG	Configuration
0x054	ADC_OFFSETSEx	RW	Offset Trimming for SE Mode
0x058	ADC_OFFSETDIFFx	RW	Offset Trimming for Diff Mode
0x05C	ADC_GAINCALx	RW	Gain Calibration
0x070	ADC_SCANFIFOCFG	RW CONFIG	Scan FIFO Configuration
0x074	ADC_SCANFIFODATA	RH(r)	Scan FIFO Read Data
0x078	ADC_SCANFIFOSTAT	RH	Scan FIFO Status
0x07C	ADC_SCANDATA	RH SYNC	Scan Data
0x080	ADC_SCANx	RW CONFIG	SCAN ENTRY
0x1000	ADC_IPVERSION_SET	R	IPVERSION
0x1004	ADC_EN_SET	RW ENABLE	Enable
0x1008	ADC_CTRL_SET	RW CONFIG	Control
0x100C	ADC_CMD_SET	W SYNC	Command
0x1010	ADC_TIMER_SET	RW CONFIG	Timer Control
0x1014	ADC_STATUS_SET	RH	Status
0x1018	ADC_MASKREQ_SET	RW SYNC	Select Channels to Be Converted
0x101C	ADC_STMASK_SET	RH SYNC	Scan Table Mask
0x1020	ADC_CMPTHR_SET	RW CONFIG	Digital Window Comparator Threshold
0x1024	ADC_IF_SET	RWH INTFLAG	Interrupt Flags
0x1028	ADC_IEN_SET	RW	Interrupt Enable
0x102C	ADC_TRIGGER_SET	RW CONFIG	Trigger
0x1050	ADC_CFGx_SET	RW CONFIG	Configuration
0x1054	ADC_OFFSETSEx_SET	RW	Offset Trimming for SE Mode
0x1058	ADC_OFFSETDIFFx_SET	RW	Offset Trimming for Diff Mode
0x105C	ADC_GAINCALx_SET	RW	Gain Calibration
0x1070	ADC_SCANFIFOCFG_SET	RW CONFIG	Scan FIFO Configuration
0x1074	ADC_SCANFIFODATA_SET	RH(r)	Scan FIFO Read Data

Offset	Name	Type	Description
0x1078	ADC_SCANFIFOSTAT_SET	RH	Scan FIFO Status
0x107C	ADC_SCANDATA_SET	RH SYNC	Scan Data
0x1080	ADC_SCANx_SET	RW CONFIG	SCAN ENTRY
0x2000	ADC_IPVERSION_CLR	R	IPVERSION
0x2004	ADC_EN_CLR	RW ENABLE	Enable
0x2008	ADC_CTRL_CLR	RW CONFIG	Control
0x200C	ADC_CMD_CLR	W SYNC	Command
0x2010	ADC_TIMER_CLR	RW CONFIG	Timer Control
0x2014	ADC_STATUS_CLR	RH	Status
0x2018	ADC_MASKREQ_CLR	RW SYNC	Select Channels to Be Converted
0x201C	ADC_STMASK_CLR	RH SYNC	Scan Table Mask
0x2020	ADC_CMPTHR_CLR	RW CONFIG	Digital Window Comparator Threshold
0x2024	ADC_IF_CLR	RWH INTFLAG	Interrupt Flags
0x2028	ADC_IEN_CLR	RW	Interrupt Enable
0x202C	ADC_TRIGGER_CLR	RW CONFIG	Trigger
0x2050	ADC_CFGx_CLR	RW CONFIG	Configuration
0x2054	ADC_OFFSETSEx_CLR	RW	Offset Trimming for SE Mode
0x2058	ADC_OFFSETDIFFx_CLR	RW	Offset Trimming for Diff Mode
0x205C	ADC_GAINCALx_CLR	RW	Gain Calibration
0x2070	ADC_SCANFIFOCFG_CLR	RW CONFIG	Scan FIFO Configuration
0x2074	ADC_SCANFIFODATA_CLR	RH(r)	Scan FIFO Read Data
0x2078	ADC_SCANFIFOSTAT_CLR	RH	Scan FIFO Status
0x207C	ADC_SCANDATA_CLR	RH SYNC	Scan Data
0x2080	ADC_SCANx_CLR	RW CONFIG	SCAN ENTRY
0x3000	ADC_IPVERSION_TGL	R	IPVERSION
0x3004	ADC_EN_TGL	RW ENABLE	Enable
0x3008	ADC_CTRL_TGL	RW CONFIG	Control
0x300C	ADC_CMD_TGL	W SYNC	Command
0x3010	ADC_TIMER_TGL	RW CONFIG	Timer Control
0x3014	ADC_STATUS_TGL	RH	Status
0x3018	ADC_MASKREQ_TGL	RW SYNC	Select Channels to Be Converted
0x301C	ADC_STMASK_TGL	RH SYNC	Scan Table Mask
0x3020	ADC_CMPTHR_TGL	RW CONFIG	Digital Window Comparator Threshold
0x3024	ADC_IF_TGL	RWH INTFLAG	Interrupt Flags
0x3028	ADC_IEN_TGL	RW	Interrupt Enable
0x302C	ADC_TRIGGER_TGL	RW CONFIG	Trigger
0x3050	ADC_CFGx_TGL	RW CONFIG	Configuration
0x3054	ADC_OFFSETSEx_TGL	RW	Offset Trimming for SE Mode
0x3058	ADC_OFFSETDIFFx_TGL	RW	Offset Trimming for Diff Mode
0x305C	ADC_GAINCALx_TGL	RW	Gain Calibration
0x3070	ADC_SCANFIFOCFG_TGL	RW CONFIG	Scan FIFO Configuration

Offset	Name	Type	Description
0x3074	ADC_SCANFIFODATA_TGL	RH(r)	Scan FIFO Read Data
0x3078	ADC_SCANFIFOSTAT_TGL	RH	Scan FIFO Status
0x307C	ADC_SCANDATA_TGL	RH SYNC	Scan Data
0x3080	ADC_SCANx_TGL	RW CONFIG	SCAN ENTRY

23.5 ADC Register Description

23.5.1 ADC_IPVERSION - IPVERSION

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

23.5.2 ADC_EN - Enable

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status** When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGS and FIFOs.
0	EN	0x0	RW	**Enable SAR Module** The EN bit enables the module. Software should write to CONFIG type registers before setting the EN bit. Software should write to SYNC type registers only after setting the EN bit.

Value	Mode	Description
0	DISABLE	Disable
1	ENABLE	Enable

23.5.3 ADC_CTRL - Control

Offset	Bit Position																															
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0						0x0			0x1F											0x0						0x0		0x0		
Access		RW						RW			RW											RW						RW		RW		
Name		HSCLKRATE						REFSEL			TIMEBASE											ADCPRESCALE						WARMUPMODE		DBGHALT		

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30:28	HSCLKRATE	0x0	RW	**High Speed Prescaler**

First level prescaler divides CLK_CMU_ADC by (HSCLKRATE + 1) to generate CLK_SRC_ADC

Value	Mode	Description
0	DIV1	Use clk_per directly.
1	DIV2	Divide clk_per by 2 before using it.
2	DIV3	Divide clk_per by 3 before using it. the resulting clk_src_adc must be 40Mhz or less
3	DIV4	Divide clk_per by 4 before using it. the resulting clk_src_adc must be 40Mhz or less
4	DIV5	Divide clk_per by 5 before using it. the resulting clk_src_adc must be 40Mhz or less
5	DIV6	Divide clk_per by 6 before using it. the resulting clk_src_adc must be 40Mhz or less
6	DIV7	Divide clk_per by 7 before using it. the resulting clk_src_adc must be 40Mhz or less
7	DIV8	Divide clk_per by 8 before using it. the resulting clk_src_adc must be 40Mhz or less

27:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24:22	REFSEL	0x0	RW	**Reference Selection**

Selects voltage reference

Value	Mode	Description
0	VREFINT	Internal voltage reference
1	VREFPL	External reference pin low voltage (1.05-1.25)
2	VREFPH	External reference pin high voltage(>1.25)
3	VDDA	Analog voltage supply
4	VREFPBUF	External reference pin buffered

21:15	TIMEBASE	0x1F	RW	**Time Base**

Number of CLK_CORE_ADC cycles (TIMEBASE + 1) needed to generate a 1 us interval for warm up and start up timing. Does not allow less than 2 cycles. A setting of 0x0 (1 cycle) is replaced with 0x1 (2 cycles).

Bit	Name	Reset	Access	Description
14:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10:5	ADCPRESCALE	0x0	RW	**Second Prescaler**
	Second level prescaler divides the CLK_SRC_ADC by (ADCPRESCALE + 1) to generate CLK_CORE_ADC and CLK_SAR			
4:3	WARMUPMODE	0x0	RW	**Warmup Mode**
	Select the warmup mode for the ADC			

Value	Mode	Description
0	NORMAL	Shut down the analog ADC after conversions have completed. Requires 5 us warmup time on first conversion of any new scan.
1	KEEPWARM	Keep ADC fully powered after conversions have completed.
2	KEEPSTANDBY	Keep ADC in low-power standby state after conversions have completed. Requires 1 us warmup time on first converion of any new scan.

2	DBGHALT	0x0	RW	**Debug Halt**
	Selects ADC behavior during debug mode			

Value	Mode	Description
0	NORMAL	Continue operation as normal during debug mode
1	HALT	Complete the current conversion and then halt during debug mode

1:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

23.5.4 ADC_CMD - Command

Offset	Bit Position																														
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset							0x0								0x0	0x0												0x0	0x0			
Access							W(nB)								W(nB)	W(nB)												W(nB)	W(nB)			
Name							SCANFIFOFLUSH								TIMERDIS	TIMEREN												SCANSTOP	SCANSTART			

Bit	Name	Reset	Access	Description
31:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25	SCANFIFOFLUSH	0x0	W(nB)	**Flush Scan FIFO**
	Flush the Scan FIFO. The ADC must be enabled, not suspended, and the clock must be running. Operation has completed when STATUS.SCANFIFOFLUSHING has gone low. The scan queue should be disabled. Any incoming scan queue data will be discarded during the flush.			
24:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	TIMERDIS	0x0	W(nB)	**Timer Disable**
	Disable the local timer and reset the counter to timer reload value			
16	TIMEREN	0x0	W(nB)	**Timer Enable**
	Enable the local timer			
15:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	SCANSTOP	0x0	W(nB)	**Scan Queue Stop**
	Stop the Scan queue. Disables Scan triggers and clears pending conversions in the Scan queue. Any conversion that has already started will continue until it is complete. If the scan queue is stopped before all entries of the scan table have completed, the remaining entries will not be converted.			
3	SCANSTART	0x0	W(nB)	**Scan Queue Start**
	Start the Scan queue. Enables triggering of the Scan queue.			
2:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

23.5.5 ADC_TIMER - Timer Control

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	RW															
Name																	TIMER															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	TIMER	0x0	RW	**Timer Period** Number of (CLK_SRC_ADC+1) cycles per timer event.

23.5.6 ADC_STATUS - Status

Offset	Bit Position																														
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0						0x0			0x0					0x0	0x0					0x0	0x0			0x0		0x0	0x0			
Access		R						R			R					R	R					R	R			R		R	R			
Name		ADCWARM						SYNCBUSY			MASKREQWRITEPENDING					TIMERACTIVE	SCANFIFOFLUSHING					SCANDATADV	SCANFIFODV			CONVERTING		SCANQUEUEPENDING	SCANQEN			

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30	ADCWARM	0x0	R	**ADC Warmed Up**
	The ADC analog front end and reference generator need warmup when coming from a powered down or standby state. This status bit indicates that the analog front end and reference are readyafter warmup.			
29:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	SYNCBUSY	0x0	R	**Synchronization in Progress**
	Indicates synchronization ongoing.			
23:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	MASKREQWRITE-PENDING	0x0	R	**MASKREQ Write pending**
	A write to MASKREQ is pending. The ADC converts using a local working mask register, and only transfers MASKREQ to the local working version when the SCAN table is not converting.			
20:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16	TIMERACTIVE	0x0	R	**Timer Active**
	The local timer is running.			
15	SCANFIFOFLUSHING	0x0	R	**Scan FIFO Flush in Progress**
	A scan data FIFO flush operation is in progress.			
14:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	SCANDATADV	0x0	R	**SCANDATA Data Valid**
	The DATA register has new data since it was last read - this is a debug bit which will clear when DATA is read in debug mode (no fifo pop).			
9	SCANFIFODV	0x0	R	**Scan FIFO Data Valid**
	There is at least one result in the SCANFIFO that is ready to be read.			
8:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	CONVERTING	0x0	R	**Ongoing Conversion**

Bit	Name	Reset	Access	Description
	The ADC is warmed up and in the process of performing a conversion.			
5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	SCANQUEUEPENDING	0x0	R	**Scan Queue Pending**
	Scan sequence is active and has not begun a conversion of any entry yet			
3	SCANQEN	0x0	R	**Scan Queued Enabled**
	Indicates that the ADC is ready to receive a trigger for conversion.			
2:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

23.5.7 ADC_MASKREQ - Select Channels to Be Converted

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	RW															
Name																	MASKREQ															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	MASKREQ	0x0	RW	**Channel Mask Request**
	Allows software to specify which channels in the scan table should be converted during a scan.			

23.5.8 ADC_STMASK - Scan Table Mask

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	R															
Name																	STMASK															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	STMASK	0x0	R	**Scan Table Mask**
	Active/working copy of the MASKREQ register that the ADC uses. It will be updated only at the end of a SCAN sequence or when MASKREQ is written and no scan is in progress.			

23.5.9 ADC_CMPTHR - Digital Window Comparator Threshold

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																0x0															
Access	RW																RW															
Name	ADGT																ADLT															

Bit	Name	Reset	Access	Description
31:16	ADGT	0x0	RW	**ADC Greater Than or Equal to Threshold**
	Compare threshold value for greater-than or equal to comparison. ADGT should be specified in a left-justified, 16-bit format regardless of the FIFO ALIGNMENT setting. Comparisons with 12-bit formats will ignore the 4 LSBs of the ADGT value. Comparisons with 8-bit formats will ignore the 8 LSBs of the ADGT value. Unipolar or bipolar mode is considered in the comparison. When ADGT is greater than ADLT, the comparison is true if the result is either greater than ADGT or less than ADLT, but false if the result falls between the values. Values should be bipolar format for differential conversions and unipolar for single ended.			
15:0	ADLT	0x0	RW	**ADC Less Than or Equal to Threshold**
	Compare threshold value for less-than or equal to comparison. ADLT should be specified in a left-justified, 16-bit format regardless of the FIFO ALIGNMENT setting. Comparisons with 12-bit formats will ignore the 4 LSBs of the ADLT value. Comparisons with 8-bit formats will ignore the 8 LSBs of the ADLT value. Unipolar or bipolar mode is considered in the comparison. When ADGT is greater than ADLT, the comparison is true if the result is either greater than ADGT or less than ADLT, but false if the result falls between the values. Values should be bipolar format for differential conversions and unipolar for single ended.			

23.5.10 ADC_IF - Interrupt Flags

Offset	Bit Position																															
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0		0x0				0x0	0x0				0x0	0x0				0x0		0x0	
Access													RW		RW				RW	RW				RW	RW				RW		RW	
Name													SCANFIFOUF		SCANFIFOOF				PORTALLOCERR	POLARITYERR				SCANTABLEDONE	SCANENTRYDONE				SCANCMP		SCANFIFODVL	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19	SCANFIFOUF	0x0	RW	**Scan FIFO Underflow**
	A scan FIFO underflow has occured.			
18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	SCANFIFOOF	0x0	RW	**Scan FIFO Overflow**
	A scan FIFO overflow has occured.			
16:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	PORTALLOCERR	0x0	RW	**Port Allocation Error**
	A pin was selected on a port which has not been allocated to the SAR12B in GPIO control.			
12	POLARITYERR	0x0	RW	**Polarity Error**
	Either two even channels or two odd channels were programmed into the channel mux selection for differential mode. In single ended mode, PORTPOS was programmed to GND without PORTNEG being GND. Results will be corrupted.			
11:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	SCANTABLEDONE	0x0	RW	**Scan Sequence Done**
	A scan table conversion completed. Set at the completion of a scan table			
7	SCANENTRYDONE	0x0	RW	**Scan Entry Done**
	A scan channel has been completed. Set at the completion of each valid entry of the scan table			
6:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	SCANCMP	0x0	RW	**Scan Result Window Compare**
	Scan digital compare window tripped.			
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	SCANFIFODVL	0x0	RW	**Scan FIFO Data Valid Level**
	A minimum of (DVL+1) entries are ready to be read from the Scan FIFO.			
0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

23.5.11 ADC_IEN - Interrupt Enable

Offset	Bit Position																															
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0		0x0				0x0	0x0				0x0	0x0				0x0		0x0	
Access													RW		RW				RW	RW				RW	RW				RW		RW	
Name													SCANFIFOUF		SCANFIFOOF				PORTALLOCERR	POLARITYERR				SCANTABLEDONE	SCANENTRYDONE				SCANCMP		SCANFIFODVL	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19	SCANFIFOUF	0x0	RW	**Scan FIFO Underflow Enable**
	Scan FIFO underflow interrupt enable			
18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	SCANFIFOOF	0x0	RW	**Scan FIFO Overflow Enable**
	Scan FIFO overflow interrupt enable			
16:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	PORTALLOCERR	0x0	RW	**Port Allocation Error Enable**
	Port Allocation Error Enable			
12	POLARITYERR	0x0	RW	**Polarity Error Enable**
	Polarity Error Enable			
11:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8	SCANTABLEDONE	0x0	RW	**Scan Sequence Done**
	Scan complete interrupt enable.			
7	SCANENTRYDONE	0x0	RW	**Scan Entry Done**
	Scan entry complete interrupt enable.			
6:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	SCANCMP	0x0	RW	**Scan Result Window Compare Enable**
	Scan Result Window Compare Enable			
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	SCANFIFODVL	0x0	RW	**Scan FIFO Data Valid Level Enable**
	Scan FIFO Data Valid Level interrupt enable			
0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

23.5.12 ADC_TRIGGER - Trigger

Offset	Bit Position																															
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x0																						0x0		0x0		
Access						RW																						RW		RW		
Name						REPDELAY																						SCANTRIGACTION		SCANTRIGSEL		

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:23	REPDELAY	0x0	RW	**delay between repetitions**

Delay value between one sequence of the table(last entry in one sequence) and the start of next sequence in continuous action mode

Value	Mode	Description
0	NODELAY	No delay
1	CYCLE2	2 conversion clock cycles
2	CYCLE4	4 conversion clock cycles
3	CYCLE8	8 conversion clock cycles
4	CYCLE16	16 conversion clock cycles
5	CYCLE32	32 conversion clock cycles
6	CYCLE64	64 conversion clock cycles
7	CYCLE128	128 conversion clock cycles
8	CYCLE256	256 conversion clock cycles
9	CYCLE512	512 conversion clock cycles
10	CYCLE1024	1024 conversion clock cycles

22:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	SCANTRIGACTION	0x0	RW	**Scan trigger action**

Enable/disable single or continuous scan sequence.

Value	Mode	Description
0	ONCE	Goes through scan table once per trigger
1	CONTINUOUS	goes through the scan table converts every entry in maskbit and keeps going through the table until stop command is issued or module is diabled

3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	SCANTRIGSEL	0x0	RW	**Scan Trigger Select**

Selects how the conversion start will be triggered

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	IMMEDIATE		Immediate triggering. The scan queue will be disabled once all conversions in the scan table are complete, unless TRIGGER-ACTION is set to continuous.
	1	TIMER		Triggers when the local timer count reaches zero.
	2	PRSCLKGRP		Triggers on PRS0 from a timer module that is using the same clock group as the ADC and has been programmed to use the same clock source as the ADC. The prescale may be different between the ADC and the timer module.
	3	PRSPOS		Triggers on asynchronous PRS0 positive edge. Requires PRS0 to go low for 3 ADCCLKs before another positive edge can be detected. Generates an additional delay of 1 to 2 CLK_SRC_ADC cycles for synchronization.
	4	PRSNEG		Triggers on asynchronous PRS0 negative edge. Requires PRS0 to go high for 3 ADCCLKs before another negative edge can be detected. Generates an additional delay of 1 to 2 CLK_SRC_ADC cycles for synchronization. PRSNEG should only be used when the PRS trigger source is from a module on a PD0 or PDHV power domain. For modules (ie: TIMER) on a PD1 power domain, PRSPOS should be used for an asynchronous trigger, and PRSCLKGRP should be used for a synchronous trigger.

23.5.13 ADC_CFGx - Configuration

Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x0												0x2			0x0										
Access						RW												RW			RW										
Name						AVERAGESEL												ANALOGGAIN			ATIME										

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:23	AVERAGESEL	0x0	RW	**Average Select**
	Number of samples to collect for averaging			

Value	Mode	Description
0	X1	1 samples for each conversion result
1	X2	2 samples for each conversion result
2	X4	4 samples for each conversion result
3	X8	8 samples for each conversion result
4	X16	16 samples for each conversion result
5	X32	32 samples for each conversion result
6	X64	64 samples for each conversion result
7	X128	128 samples for each conversion result
8	X256	256 samples for each conversion result
9	X512	512 samples for each conversion result
10	X1024	1024 samples for each conversion result

Bit	Name	Reset	Access	Description
22:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14:12	ANALOGGAIN	0x2	RW	**Analog Gain**
	Select the analog gain for the ADC.			

Value	Mode	Description
0	GAIN_0_3125	Gain for sampling cap of 0.3125x
1	GAIN_0_5	Gain for sampling cap of 0.5x
2	GAIN_1	Gain for sampling cap of 1x
3	GAIN_2	Gain for sampling cap of 2x
4	GAIN_4	Gain for sampling cap of 4x

Bit	Name	Reset	Access	Description
11:2	ATIME	0x0	RW	**Acquisition Time**
	Select the acquisition time for each sample in CLK_CORE_ADC clocks. Acquisition time is equal to (ATIME + 1) CLK_CORE_ADC clocks.			
1:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

23.5.14 ADC_OFFSETSEx - Offset Trimming for SE Mode

Offset	Bit Position																													
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x800																0x800											
Access					RW																RW											
Name					SENEG																SEPOS											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:16	SENEG	0x800	RW	**Offset Trim Single Ended Negative**
	Used for the offset calibration of the ADC for single ended negative conversions			
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:0	SEPOS	0x800	RW	**Offset Trim Single Ended Positive**
	Used for the offset calibration of the ADC for single ended positive conversions			

23.5.15 ADC_OFFSETDIFFx - Offset Trimming for Diff Mode

Offset	Bit Position																													
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x800											
Access																					RW											
Name																					DIFF											

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:0	DIFF	0x800	RW	**Offset Trim Differential**
	Offset calibration used for differential mode conversions.			

23.5.16 ADC_GAINCALx - Gain Calibration

Offset	Bit Position																														
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset						0x60										0x60										0x60						
Access						RW										RW										RW						
Name						DIFF										SENEG										SEPOS						

Bit	Name	Reset	Access	Description
31:27	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
26:20	DIFF	0x60	RW	**Gain Calibration for Differential**
	Gain calibration for differential conversions.			
19:17	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
16:10	SENEG	0x60	RW	**Gain Calibration for Single Ended Negative**
	Gain calibration for single ended negative conversions.			
9:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:0	SEPOS	0x60	RW	**Gain Calibration for Single Ended Positive**
	Gain calibration for single ended positive conversions.			

23.5.17 ADC_SCANFIFOCFG - Scan FIFO Configuration

Offset	Bit Position																															
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x3				0x0	0x0		
Access																									RW				RW	RW		
Name																									DVL				SHOWID	ALIGNMENT		

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:4	DVL	0x3	RW	**Data Valid Level**

Data valid level before requesting DMA transfer. If the number of words in the FIFO reaches or exceeds DVL+1, DMA requests will be generated.

Value	Mode	Description
0	VALID1	When 1 entry in the scan FIFO is valid, set the SCANFIFODVL interrupt and request DMA.
1	VALID2	When 2 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
2	VALID3	When 3 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
3	VALID4	When 4 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
4	VALID5	When 5 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
5	VALID6	When 6 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
6	VALID7	When 7 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
7	VALID8	When 8 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
8	VALID9	When 9 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
9	VALID10	When 10 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
10	VALID11	When 11 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
11	VALID12	When 12 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
12	VALID13	When 13 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
13	VALID14	When 14 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
14	VALID15	When 15 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.

Bit	Name	Reset	Access	Description
	15	VALID16		When 16 entries in the scan FIFO are valid, set the SCANFIFODVL interrupt and request DMA.
3	SHOWID	0x0	RW	**Show ID**
	Enable ID in output words.			
2:0	ALIGNMENT	0x0	RW	**Alignment**
	Alignment of output data written into FIFO.			

Value	Mode	Description
0	RIGHT16	ID[7:0], SIGN_EXT, DATA[15:0]
1	RIGHT12	ID[7:0], SIGN_EXT, DATA[11:0]
2	RIGHT8	ID[7:0], SIGN_EXT, DATA[7:0]
3	LEFT16	DATA[15:0], 00000000, ID[7:0]
4	LEFT12	DATA[11:0], 000000000000, ID[7:0]
5	LEFT8	DATA[7:0], 0000000000000000, ID[7:0]

23.5.18 ADC_SCANFIFODATA - Scan FIFO Read Data

Offset	Bit Position																													
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R(r)																															
Name	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	R(r)	**Data**
	Reads and pops the oldest value from the scan FIFO.			

23.5.19 ADC_SCANFIFOSTAT - Scan FIFO Status

Offset	Bit Position																															
0x078	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0				
Access																												R				
Name																												FIFOREADCNT				

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4:0	FIFOREADCNT	0x0	R	**FIFO Read Count** Number of valid entries available to read.

23.5.20 ADC_SCANDATA - Scan Data

Offset	Bit Position																															
0x07C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	DATA																															

Bit	Name	Reset	Access	Description
31:0	DATA	0x0	R	**Data** Reads the most recent data word from the scan FIFO, but does not pop a value. Even if the FIFO has overflowed and stopped updating, the most recent conversion will continue to overwrite SCANDATA.

23.5.21 ADC_SCANx - SCAN ENTRY

Offset	Bit Position																															
0x080	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	0x0	0x0				0x0				0x0				0x0			
Access															RW	RW	RW				RW				RW				RW			
Name															CMP	CFG	PORTPOS				PINPOS				PORTNEG				PINNEG			

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	CMP	0x0	RW	**Comparison Enable**
	Enable digital window comparison for this entry.			
16	CFG	0x0	RW	**Configuration Group Select**
	Select which configuration group (CFGx register) is used with this entry.			

Value	Mode	Description
0	CONFIG0	Use configuration group 0
1	CONFIG1	Use configuration group 1

Bit	Name	Reset	Access	Description
15:12	PORTPOS	0x0	RW	**Positive Port Select**
	Port (A, B, C, D) or special signal assigned to the positve input of the ADC			

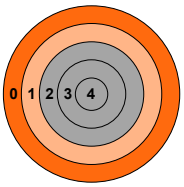
Value	Mode	Description
0	GND	Ground
1	SUPPLY	Supply Pin - Select specific supply using PINPOS
8	PORTA	Port A - Select pin number using PINPOS
9	PORTB	Port B - Select pin number using PINPOS
10	PORTC	Port C - Select pin number using PINPOS
11	PORTD	Port D - Select pin number using PINPOS

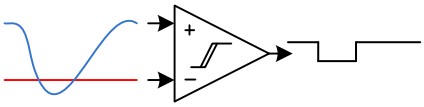
Bit	Name	Reset	Access	Description
11:8	PINPOS	0x0	RW	**Positive Pin Select**
	Pin number for the positive input of the ADC.			
7:4	PORTNEG	0x0	RW	**Negative Port Select**
	Port (A, B, C, D) or special signal assigned to the negative input of the ADC			

Value	Mode	Description
0	GND	Ground (single-ended)
8	PORTA	Port A - Select pin number using PINNEG
9	PORTB	Port B - Select pin number using PINNEG
10	PORTC	Port C - Select pin number using PINNEG
11	PORTD	Port D - Select pin number using PINNEG

Bit	Name	Reset	Access	Description
3:0	PINNEG	0x0	RW	**Negative Pin Select**
	Pin number for the negative input of the ADC.			

24. ACMP - Analog Comparator





Quick Facts

What?

The ACMP compares two analog signals and returns a digital value telling which is greater.

Why?

Applications often do not need to know the exact value of an analog signal, only if it has passed a certain threshold. Often the voltage must be monitored continuously, which requires extremely low power consumption.

How?

Available down to Energy Mode EM1, the ACMP can wake up the system when input signals pass the threshold. The analog comparator can compare two analog signals or one analog signal and a highly configurable internal reference.

24.1 Introduction

The Analog Comparator compares the voltage of two analog inputs and outputs a digital signal indicating which input voltage is higher. Inputs can either be from internal references or from external pins. Response time, and thereby the current consumption, can be configured by altering the current supply to the comparator.

24.2 Features

- Internal and external input selections:
  - External port I/O routed via ABUS
  - Internal 1.25 V bandgap reference voltage with programmable divider
  - Internal 2.5 V bandgap reference voltage with programmable divider
  - AVDD supply voltage with programmable divider
- Voltage supply monitoring
- Configurable hysteresis
- Selectable response time
- Operational in EM0 and EM1
- Asynchronous interrupt generation on selectable edges
- Configurable output state when inactive
- Output routing options:
  - Route to GPIO via DBUS
  - Route to most peripherals via PRS

24.3 Functional Description

An overview of the ACMP is shown in Figure 24.1. ACMP Overview.

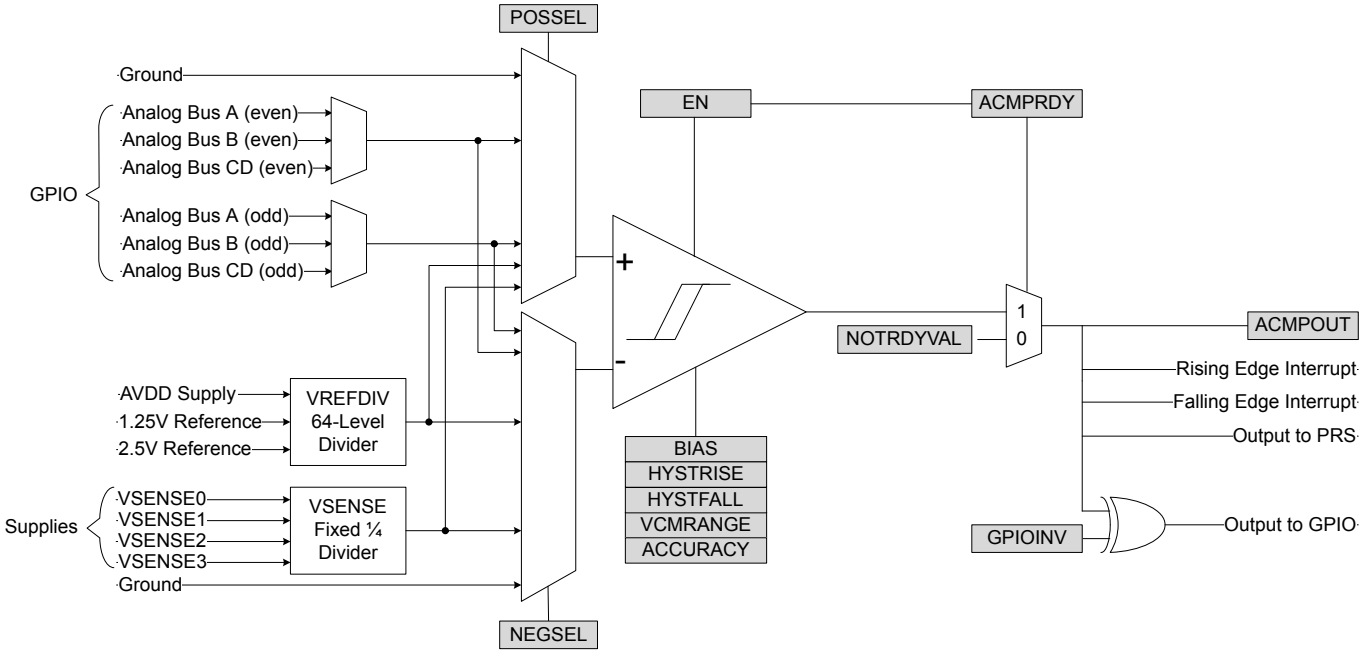


Figure 24.1. ACMP Overview

The comparator has two analog inputs: one positive and one negative. When the comparator is active, the output indicates which of the two input voltages is higher. When the voltage on the positive input is higher than the voltage on the negative input, the digital output is high and vice versa.

In addition to the comparator core, the ACMP front-end includes reference sources, voltage divider circuits, and input muxes to route signals to the positive and negative inputs. The output from the ACMP is observable in the ACMP_STATUS register. The digital output may also be routed to a GPIO via ACMPn.DIGOUT, and used as a PRS producer by setting SOURCESEL to ACMPn and SIGSEL to OUT.

24.3.1 Configuration and Control

The ACMP is configured and controlled through three registers: ACMP_CFG, ACMP_CTRL, and ACMP_INPUTCTRL. Configuration through ACMP_CFG needs to happen before the ACMP is enabled.The control registers ACMP_CTRL and ACMP_INPUTCTRL can only be updated after the ACMP is enabled. The ACMP is enabled by setting the EN bit in ACMP_EN. If ACMP_CFG is updated when EN = 1, or ACMP_CTRL / ACMP_INPUTCTRL is updated while EN = 0, a bus fault is issued.

The input muxes are configured in the POSSEL / NEGSEL bitfields in ACMP_INPUTCTRL. All references and inputs are available in the modes defined for these two registers. The INPUTCTRL bit in ACMP_SYNCBUSY should be checked before writing to ACMP_INPUTCTRL. If the ACMP_SYNCBUSY.INPUTCTRL bit is 1, it means a previous write to the ACMP_INPUTCTRL register is pending, and software should wait until ACMP_SYNCBUSY.INPUTCTRL bit reads 0.

The POSSEL and NEGSEL muxes share several resources on the device, such as the VREFDIV and VSENSE divider circuits, and ABUS routing capabilities. Thus, there are some constraints on the POSSEL / NEGSEL configurations:

- POSSEL and NEGSEL cannot select an even numbered GPIO pin at the same time.
- POSSEL and NEGSEL cannot select an odd numbered GPIO pin at the same time.
- POSSEL and NEGSEL cannot both select a supply voltage via one of the VSENSE inputs.
- POSSEL and NEGSEL cannot both select an input using VREFDIV.
- If POSSEL = EXTPx, a low power reference (postfixed with LP) cannot be selected for NEGSEL.

If one of these constraints are violated, the INPUTCONFLICT status flag and INPUTCONFLICTIF interrupt flag will be set.

The ACMP also uses shared chip-level analog bus resources to connect to external GPIO pins. Which bus the ACMP is using depends on the configuration of POSSEL and NEGSEL. To allow the ACMP to control an analog bus, the bus must be allocated to ACMP in the GPIO module, using the GPIO_xBUSALLOC registers. For example, pin PB5 is an odd-numbered pin on port PB, and could connect via either analog bus BODD0 or BODD1. This is configured using the BODD0 or BODD1 field in GPIO_BBUSALLOC. If both inputs to the ACMP are routed to GPIO pins through ABUS, one input will use an ODD bus (odd-numbered GPIO pins) and the other will use an EVEN bus (even-numbered GPIO pins).

If the ACMP peripheral is trying to access a bus that has not been allocated to that instance of ACMP, the PORTALLOCERR status flag and PORTALLOCERRIF interrupt flag will be set.

24.3.2 Warmup Time

When the comparator is enabled or the input muxes are reconfigured, it requires some time to stabilize. On first enable (ACMP_EN.EN = 1), the comparator core requires 2.5 us to stabilize. In addition to this, any references selected may require some time to warm up. See table Table 24.1. Warmup Time for warmup times for the different references. When reconfiguring the ACMP (without disabling it), only the warmup times given in the table will be observed. When the comparator is ready for use, the ACMPRDY status flag and the ACMPRDY interrupt flag will be set.

During the warmup time and when the comparator is inactive, the comparator output will be set to the state defined by the NOTRDYVAL bit in ACMP_CTRL.

Table 24.1. Warmup Time

Reference	Warmup time
Low power reference: POSSEL / NEGSEL = *LP	100 us
VSENSE: POSSEL / NEGSEL = VSENSE*	7 us
VREF: POSSEL / NEGSEL = VREF*	2 us
None of the above	0.5 us

24.3.3 Response Time

There is a delay from when the input voltage changes polarity to when the output toggles. This delay is called the response time and can be altered by increasing or decreasing the bias current to the comparator through the BIASPROG bitfield in the ACMP_CFG register. The current and speed of the circuit increase as the value of BIASPROG is increased. See the part data sheet for specific current and response times releated to setting of BIASPROG.

24.3.4 Hysteresis

When the hysteresis level is set to a non-zero value, the digital output will not toggle until the positive input voltage is at a voltage equal to the hysteresis level above or below the negative input voltage (see Figure 24.2. Hysteresis). This feature can be used avoid continual comparator output changes due to input noise when the positive and negative inputs are similar. Hysteresis requires the input difference to exceed the hysteresis threshold before the output can change and can reject limited amounts of noise. The hystersis in ACMP can be configured to three different levels (10 mV, 20 mV, 30 mV), and can be enabled on rising, falling, or both edges. The rising and falling edge hysteresis values are independent, and can be set to different magnitudes depending on the needs of the application. Hysteresis is controlled for rising and falling edges via the ACMP_CFG.HYSTRISE and ACMP_CFG.HYSTFALL bitfields, respectively.

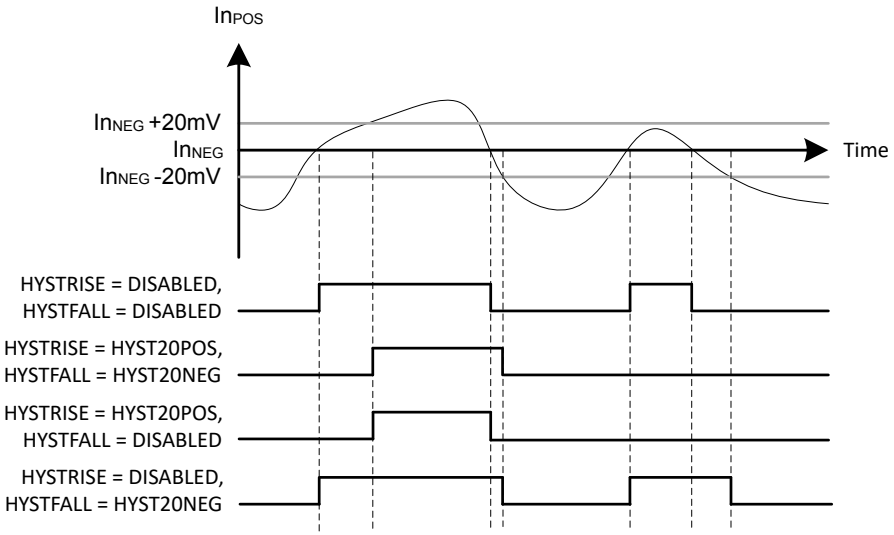


Figure 24.2. Hysteresis

24.3.5 VREFDIV Sources

The ACMP has two internal bandgap references: 2.5 V and 1.25 V. In addition, AVDD can be used as a reference. To select one of these references, configure POSSEL / NEGSEL to VREFDIVAVDD, VREFDIV1V25, or VREFDIV2V5. The ACMP also includes sample/hold functionality to reduce energy consumption. To enable the sample/hold feature, select VREFDIVAVDDLP, VREFDIV1V25LP, or VREFDIV2V5LP. These references can be divided by configuring VREFDIV in ACMP_INPUTCTRL. This division factor will be VREFDIV / 63, such that VREFOUT = VREFIN * (VREFDIV / 63).

24.3.6 Supply Voltage Monitoring (VSENSE)

The ACMP can be used to monitor supply voltages. This is done by selecting VSENSE01DIV4(LP) or VSENSE11DIV4(LP) for either POSSEL or NEGSEL. Note that the input to the comparator core will be divided by 4, as illustrated in Figure 24.1. ACMP Overview. To reduce energy consumption, a sample/hold circuit can be used to periodically sample the power supplies. To enable this, select VSENSE01DIV4LP or VSENSE11DIV4LP in POSSEL / NEGSEL. Because the sample/hold feature uses the comparator in a non-continuous fashion, enabling this will increase response times and reduce the accuracy of the comparator. The connections between VSENSE0 and VSENSE1 to power supplies are summarized in the table below.

Table 24.2. VSENSE connections

ACMP instance	VSENSE0	VSENSE1	VSENSE2	VSENSE3
ACMP0	AVDD	IOVDD	(reserved)	(reserved)
ACMP1	DVDD	(reserved)	(reserved)	(reserved)

24.3.7 Input Range, Common Mode, and Accuracy Settings

Both inputs to ACMP have an input voltage range from 0 to AVDD. When routed through the ABUS to a GPIO pin, the maximum voltage on an input will be limited by the lower of the AVDD or GPIO supply voltage for that I/O pin (typically IOVDD).

The common-mode input voltage is the average of the voltages seen at the two ACMP inputs. By default the ACMP_CFG.VCMRANGE bitfield is configured to FULL, and the common mode range is from 0 to AVDD. This is the recommended setting for most situations.

When ACMP_CFG.BIAS is configured for a lower bias level (0, 1, 2, or 3), there may be additional offset if the common-mode voltage is above 3.0 V. In a system where BIAS settings 0-3 are used and the common-mode voltage will be above 3.0 V, the comparator offset can be improved by setting ACMP_CFG.VCMRANGE to HIGH.

The ACMP also has an adjustable accuracy setting (ACCURACY in ACMP_CFG). ACCURACY is set to LOW by default, which conserves power, but may have degraded performance for rapidly changing analog Port selections in either the ACMP or the GPIO. ACCURACY can be set to HIGH to ensure ACMP accuracy (at the expense of extra power consumption), when configuration changes are expected at a high rate.

24.3.8 Interrupts and PRS Output

The analog comparator includes independent output flags for rising edge (RISEIF) and falling edge (FALLIF) events. These will be set when a rising or falling edge is detected, respectively.

Three other interrupt sources are also available. PORTALLOCERRIF and INPUTCONFLICTIF are input configuration error flags, detailed in 24.3.1 Configuration and Control. The ACMPRDYIF flag indicates comparator stability after the warmup period.

The comparator output is available as an asynchronous PRS producer, and can be routed to other peripherals in the system via PRS.

24.3.9 Output to GPIO

The output from the comparator is available as alternate functions to the GPIO pins. Each pin connection can be enabled/disabled separately using the GPIO module control registers. See the device data sheet for the available locations for each signal.

The GPIO pin must also be set as output. The output to the GPIO can be inverted by setting the GPIOINV bit in ACMP_CTRL.

24.4 ACMP Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	ACMP_IPVERSION	R	IP Version ID
0x004	ACMP_EN	RW ENABLE	ACMP Enable
0x008	ACMP_SWRST	RW SWRST	Software Reset
0x00C	ACMP_CFG	RW CONFIG	Configuration Register
0x010	ACMP_CTRL	RW	Control Register
0x014	ACMP_INPUTCTRL	RW SYNC	Input Control Register
0x018	ACMP_STATUS	RH	Status Register
0x01C	ACMP_IF	RWH INTFLAG	Interrupt Flag Register
0x020	ACMP_IEN	RW	Interrupt Enable Register
0x024	ACMP_SYNCBUSY	RH	Syncbusy
0x1000	ACMP_IPVERSION_SET	R	IP Version ID
0x1004	ACMP_EN_SET	RW ENABLE	ACMP Enable
0x1008	ACMP_SWRST_SET	RW SWRST	Software Reset
0x100C	ACMP_CFG_SET	RW CONFIG	Configuration Register
0x1010	ACMP_CTRL_SET	RW	Control Register
0x1014	ACMP_INPUTCTRL_SET	RW SYNC	Input Control Register
0x1018	ACMP_STATUS_SET	RH	Status Register
0x101C	ACMP_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1020	ACMP_IEN_SET	RW	Interrupt Enable Register
0x1024	ACMP_SYNCBUSY_SET	RH	Syncbusy
0x2000	ACMP_IPVERSION_CLR	R	IP Version ID
0x2004	ACMP_EN_CLR	RW ENABLE	ACMP Enable
0x2008	ACMP_SWRST_CLR	RW SWRST	Software Reset
0x200C	ACMP_CFG_CLR	RW CONFIG	Configuration Register
0x2010	ACMP_CTRL_CLR	RW	Control Register
0x2014	ACMP_INPUTCTRL_CLR	RW SYNC	Input Control Register
0x2018	ACMP_STATUS_CLR	RH	Status Register
0x201C	ACMP_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2020	ACMP_IEN_CLR	RW	Interrupt Enable Register
0x2024	ACMP_SYNCBUSY_CLR	RH	Syncbusy
0x3000	ACMP_IPVERSION_TGL	R	IP Version ID
0x3004	ACMP_EN_TGL	RW ENABLE	ACMP Enable
0x3008	ACMP_SWRST_TGL	RW SWRST	Software Reset
0x300C	ACMP_CFG_TGL	RW CONFIG	Configuration Register
0x3010	ACMP_CTRL_TGL	RW	Control Register
0x3014	ACMP_INPUTCTRL_TGL	RW SYNC	Input Control Register
0x3018	ACMP_STATUS_TGL	RH	Status Register
0x301C	ACMP_IF_TGL	RWH INTFLAG	Interrupt Flag Register

Offset	Name	Type	Description
0x3020	ACMP_IEN_TGL	RW	Interrupt Enable Register
0x3024	ACMP_SYNCBUSY_TGL	RH	Syncbusy

24.5 ACMP Register Description

24.5.1 ACMP_IPVERSION - IP Version ID

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x5																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x5	R	**IP version ID**

24.5.2 ACMP_EN - ACMP Enable

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status** When En is cleared, DISABLING is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFOs.
0	EN	0x0	RW	**Module enable** The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.

24.5.3 ACMP_SWRST - Software Reset

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Software reset busy status**
0	SWRST	0x0	W	**Software reset**

24.5.4 ACMP_CFG - Configuration Register

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset															0x0	0x0					0x0		0x0							0x4		
Access															RW	RW					RW		RW							RW		
Name															ACCURACY	VCMRANGE					HYSTFALL		HYSTRISE							BIAS		

Bit	Name	Reset	Access	Description
31:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17	ACCURACY	0x0	RW	**ACMP accuracy mode**

Setting ACCURACY to HIGH reduces the noise in the signal input path of the ACMP. Note, high frequency changes can cause the ACMP performance to degrade. For such uses, such as quickly scanning through multiple channels or setting the ACMP to oscillate for capacitive sense, this should be set to HIGH.

Value	Mode	Description
0	LOW	ACMP operates in low-accuracy mode but consumes less current.
1	HIGH	ACMP operates in high-accuracy mode but consumes more current.

Bit	Name	Reset	Access	Description
16	VCMRANGE	0x0	RW	**Input Range**

Adjust performance of the comparator for a given input voltage range.

Value	Mode	Description
0	FULL	Input common mode voltage can be rail-to-rail. This setting can always be used for BIAS >= 4. When BIAS is 0, 1, 2, or 3, additional offset may occur at common mode voltages above 3.0 V.
1	HIGH	Input common mode voltage above 3.0 V. This setting can be used when BIAS is 0, 1, 2, or 3 to improve the offset performance at common mode voltages above 3.0 V.

Bit	Name	Reset	Access	Description
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:10	HYSTFALL	0x0	RW	**Hysteresis mode for falling edge**

Set hysteresis mode

Value	Mode	Description
0	DISABLED	Hysteresis mode disabled on negative edge transitions
1	HYST10NEG	10mV Hysteresis on negative edge transitions
2	HYST20NEG	20mV hysteresis on negative edge transitions
3	HYST30NEG	30mV hysteresis on negative edge transitions

Bit	Name	Reset	Access	Description
9:8	HYSTRISE	0x0	RW	**Hysteresis mode for rising edge**

Set hysteresis mode

Value	Mode	Description

Bit	Name	Reset	Access	Description
	0	DISABLED		hysteresis disabled on positive edge transition
	1	HYST10POS		10mV hysteresis on positive edge transitions
	2	HYST20POS		20mV hysteresis on positive edge transitions
	3	HYST30POS		30mV hysteresis on positive edge transitions
7:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	BIAS	0x4	RW	**Bias Configuration**
	These bits control the bias current level. See the data sheet for details.			

24.5.5 ACMP_CTRL - Control Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															GPIOINV	NOTRDYVAL

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	GPIOINV	0x0	RW	**Comparator GPIO Output Invert**
	Set this bit to 1 to invert the comparator alternate function output to GPIO.			
	Value	Mode		Description
	0	NOTINV		The comparator output to GPIO is not inverted
	1	INV		The comparator output to GPIO is inverted
0	NOTRDYVAL	0x0	RW	**Not Ready Value**
	The value of this bit is used as the comparator output when the comparator is not ready.			
	Value	Mode		Description
	0	LOW		ACMP output is 0 when the ACMP is not ready.
	1	HIGH		ACMP output is 1 when the ACMP is not ready.

24.5.6 ACMP_INPUTCTRL - Input Control Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0						0x0								0x0							
Access											RW						RW								RW							
Name											VREFDIV						NEGSEL								POSSEL							

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:16	VREFDIV	0x0	RW	**VREF division**
	Set division factor for VREFDIV. VREFOUT = VREFIN * (VREFDIV / 63)			
15:8	NEGSEL	0x0	RW	**Negative Input Select**
	Select negative input.			

Value	Mode	Description
0	VSS	VSS
16	VREFDIVAVDD	Divided AVDD
17	VREFDIVAVDDLP	Low-Power Divided AVDD
18	VREFDIV1V25	Divided 1V25 reference
19	VREFDIV1V25LP	Low-power Divided 1V25 reference
20	VREFDIV2V5	Divided 2V5 reference
21	VREFDIV2V5LP	Low-power Divided 2V5 reference
32	VSENSE01DIV4	VSENSE0 divided by 4
33	VSENSE01DIV4LP	Low-power VSENSE0 divided by 4
34	VSENSE11DIV4	VSENSE1 divided by 4
35	VSENSE11DIV4LP	Low-power VSENSE1 divided by 4
36	VSENSE21DIV4	VSENSE2 divided by 4
37	VSENSE21DIV4LP	Low-power VSENSE2 divided by 4
38	VSENSE31DIV4	VSENSE3 divided by 4
39	VSENSE31DIV4LP	Low-power VSENSE3 divided by 4
128	PA0	Port A, Pin0
129	PA1	Port A, Pin1
130	PA2	Port A, Pin2
131	PA3	Port A, Pin3
132	PA4	Port A, Pin4
133	PA5	Port A, Pin5
134	PA6	Port A, Pin6
135	PA7	Port A, Pin7
136	PA8	Port A, Pin8
137	PA9	Port A, Pin9

Bit	Name	Reset	Access	Description
	138	PA10		Port A, Pin10
	139	PA11		Port A, Pin11
	140	PA12		Port A, Pin12
	141	PA13		Port A, Pin13
	142	PA14		Port A, Pin14
	143	PA15		Port A, Pin15
	144	PB0		Port B, Pin0
	145	PB1		Port B, Pin1
	146	PB2		Port B, Pin2
	147	PB3		Port B, Pin3
	148	PB4		Port B, Pin4
	149	PB5		Port B, Pin5
	150	PB6		Port B, Pin6
	151	PB7		Port B, Pin7
	152	PB8		Port B, Pin8
	153	PB9		Port B, Pin9
	154	PB10		Port B, Pin10
	155	PB11		Port B, Pin11
	156	PB12		Port B, Pin12
	157	PB13		Port B, Pin13
	158	PB14		Port B, Pin14
	159	PB15		Port B, Pin15
	160	PC0		Port C, Pin0
	161	PC1		Port C, Pin1
	162	PC2		Port C, Pin2
	163	PC3		Port C, Pin3
	164	PC4		Port C, Pin4
	165	PC5		Port C, Pin5
	166	PC6		Port C, Pin6
	167	PC7		Port C, Pin7
	168	PC8		Port C, Pin8
	169	PC9		Port C, Pin9
	170	PC10		Port C, Pin10
	171	PC11		Port C, Pin11
	172	PC12		Port C, Pin12
	173	PC13		Port C, Pin13
	174	PC14		Port C, Pin14
	175	PC15		Port C, Pin15
	176	PD0		Port D, Pin0
	177	PD1		Port D, Pin1
	178	PD2		Port D, Pin2

Bit	Name	Reset	Access	Description
	179	PD3		Port D, Pin3
	180	PD4		Port D, Pin4
	181	PD5		Port D, Pin5
	182	PD6		Port D, Pin6
	183	PD7		Port D, Pin7
	184	PD8		Port D, Pin8
	185	PD9		Port D, Pin9
	186	PD10		Port D, Pin10
	187	PD11		Port D, Pin11
	188	PD12		Port D, Pin12
	189	PD13		Port D, Pin13
	190	PD14		Port D, Pin14
	191	PD15		Port D, Pin15
7:0	POSSEL	0x0	RW	**Positive Input Select**
	Select positive input.			

Value	Mode	Description
0	VSS	VSS
16	VREFDIVAVDD	Divided AVDD
17	VREFDIVAVDDLP	Low-Power Divided AVDD
18	VREFDIV1V25	Divided 1V25 reference
19	VREFDIV1V25LP	Low-power Divided 1V25 reference
20	VREFDIV2V5	Divided 2V5 reference
21	VREFDIV2V5LP	Low-power Divided 2V5 reference
32	VSENSE01DIV4	VSENSE0 divided by 4
33	VSENSE01DIV4LP	Low-power VSENSE0 divided by 4
34	VSENSE11DIV4	VSENSE1 divided by 4
35	VSENSE11DIV4LP	Low-power VSENSE1 divided by 4
36	VSENSE21DIV4	VSENSE2 divided by 4
37	VSENSE21DIV4LP	Low-power VSENSE2 divided by 4
38	VSENSE31DIV4	VSENSE3 divided by 4
39	VSENSE31DIV4LP	Low-power VSENSE3 divided by 4
80	EXTPA	External interface, base is PA0.
81	EXTPB	External interface, base is PB0.
82	EXTPC	External interface, base is PC0.
83	EXTPD	External interface, base is PD0.
128	PA0	Port A, Pin0
129	PA1	Port A, Pin1
130	PA2	Port A, Pin2
131	PA3	Port A, Pin3
132	PA4	Port A, Pin4

Bit	Name	Reset	Access	Description
	133	PA5		Port A, Pin5
	134	PA6		Port A, Pin6
	135	PA7		Port A, Pin7
	136	PA8		Port A, Pin8
	137	PA9		Port A, Pin9
	138	PA10		Port A, Pin10
	139	PA11		Port A, Pin11
	140	PA12		Port A, Pin12
	141	PA13		Port A, Pin13
	142	PA14		Port A, Pin14
	143	PA15		Port A, Pin15
	144	PB0		Port B, Pin0
	145	PB1		Port B, Pin1
	146	PB2		Port B, Pin2
	147	PB3		Port B, Pin3
	148	PB4		Port B, Pin4
	149	PB5		Port B, Pin5
	150	PB6		Port B, Pin6
	151	PB7		Port B, Pin7
	152	PB8		Port B, Pin8
	153	PB9		Port B, Pin9
	154	PB10		Port B, Pin10
	155	PB11		Port B, Pin11
	156	PB12		Port B, Pin12
	157	PB13		Port B, Pin13
	158	PB14		Port B, Pin14
	159	PB15		Port B, Pin15
	160	PC0		Port C, Pin0
	161	PC1		Port C, Pin1
	162	PC2		Port C, Pin2
	163	PC3		Port C, Pin3
	164	PC4		Port C, Pin4
	165	PC5		Port C, Pin5
	166	PC6		Port C, Pin6
	167	PC7		Port C, Pin7
	168	PC8		Port C, Pin8
	169	PC9		Port C, Pin9
	170	PC10		Port C, Pin10
	171	PC11		Port C, Pin11
	172	PC12		Port C, Pin12
	173	PC13		Port C, Pin13

Bit	Name	Reset	Access	Description
	174	PC14		Port C, Pin14
	175	PC15		Port C, Pin15
	176	PD0		Port D, Pin0
	177	PD1		Port D, Pin1
	178	PD2		Port D, Pin2
	179	PD3		Port D, Pin3
	180	PD4		Port D, Pin4
	181	PD5		Port D, Pin5
	182	PD6		Port D, Pin6
	183	PD7		Port D, Pin7
	184	PD8		Port D, Pin8
	185	PD9		Port D, Pin9
	186	PD10		Port D, Pin10
	187	PD11		Port D, Pin11
	188	PD12		Port D, Pin12
	189	PD13		Port D, Pin13
	190	PD14		Port D, Pin14
	191	PD15		Port D, Pin15

24.5.7 ACMP_STATUS - Status Register

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0		0x0
Access																												R	R	R		R
Name																												PORTALLOCERR	INPUTCONFLICT	ACMPRDY		ACMPOUT

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PORTALLOCERR	0x0	R	**Port allocation error**
	The port selected by INPUTCTRL_POSSEL or INPUTCTRL_NEGSEL is not allocated to this ACMP. Port allocation needs to be configured in the GPIO module.			
3	INPUTCONFLICT	0x0	R	**INPUT conflict**
	INPUTCTRL_POSSEL and INPUTCTRL_NEGSEL is configured illegally. See section XX for restrictions.			
2	ACMPRDY	0x0	R	**Analog Comparator Ready**
	Analog comparator ready status.			
1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	ACMPOUT	0x0	R	**Analog Comparator Output**
	Analog comparator output value.			

24.5.8 ACMP_IF - Interrupt Flag Register

Offset	Bit Position																														
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												PORTALLOCERR	INPUTCONFLICT	ACMPRDY	FALL	RISE

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PORTALLOCERR	0x0	RW	**ABUS allocation error**
	The port selected by INPUTCTRL_POSSEL or INPUTCTRL_NEGSEL is not allocated to this ACMP. Port allocation needs to be configured in the GPIO module.			
3	INPUTCONFLICT	0x0	RW	**Input conflict**
	INPUTCTRL_POSSEL and INPUTCTRL_NEGSEL is configured illegally.			
2	ACMPRDY	0x0	RW	**ACMP ready Interrupt flag**
	Indicates that the analog comparator is ready and references have settled. Note at lower bias settings (BIAS<=3), the ACMPRDY bit may not be reliable and additional software wait may be required. Refer to "Warmup Time" section for additional information.			
1	FALL	0x0	RW	**Falling Edge Triggered Interrupt Flag**
	Indicates that there has been a falling edge on the analog comparator output.			
0	RISE	0x0	RW	**Rising Edge Triggered Interrupt Flag**
	Indicates that there has been a rising edge on the analog comparator output.			

24.5.9 ACMP_IEN - Interrupt Enable Register

Offset	Bit Position																														
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												PORTALLOCERR	INPUTCONFLICT	ACMPRDY	FALL	RISE

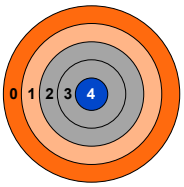
Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PORTALLOCERR	0x0	RW	**ABUS allocation error interrupt enable**
3	INPUTCONFLICT	0x0	RW	**Input conflict interrupt enable**
2	ACMPRDY	0x0	RW	**ACMP ready interrupt enable**
1	FALL	0x0	RW	**Falling edge interrupt enable**
0	RISE	0x0	RW	**Rising edge interrupt enable**

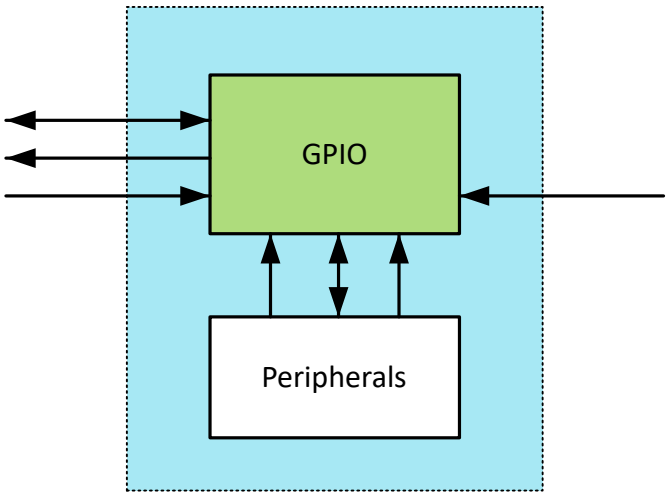
24.5.10 ACMP_SYNCBUSY - Syncbusy

Offset	Bit Position																														
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																INPUTCTRL

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	INPUTCTRL	0x0	R	**Syncbusy for INPUTCTRL**
	Synchronization of INPUTCTRL ongoing			

25. GPIO - General Purpose Input/Output





Quick Facts

What?

The GPIO block is used for pin configurations as well as routing for peripheral pin connections.

Why?

User-friendly and highly configurable input/output pins are essential for supporting a wide range of communication protocols and minimizing the need for complex software control. Additionally, flexible routing of peripheral functions simplifies PCB layout design.

How?

Each pin on the device can be individually configured as either an input or an output with several different drive modes. Also, individual bit manipulation registers minimizes control overhead. Peripheral connections to pins can be routed to pins as desired, solving congestion and contention issues that may arise with limited routing flexibility. Fully asynchronous interrupts can also be generated from any pin.

25.1 Introduction

In the SiXG301 devices the General Purpose Input/Output (GPIO) pins are organized into ports with up to 16 pins each. These GPIO pins can be individually configured as either an output or input. More advanced configurations like open-drain, open-source, and glitch filtering can be configured for each individual GPIO pin. Peripheral resources, like Timer PWM outputs or USART RX/TX can be routed to the GPIO pins as desired by the user. Finally, the input value of a pin can be routed through the Peripheral Reflex System to other peripherals or used to trigger an external interrupt.

25.2 Features

- Individual configuration for each pin
  - Tristate (reset state)
  - Push-pull
  - Open-drain
  - Pull-up resistor
  - Pull-down resistor
  - Programable Slewrate Control
- EM4 IO pin retention
  - Output enable
  - Output value
  - Pull enable
  - Pull direction
- EM4 wake-up on selected GPIO pins
- Glitch suppression input filter
- Extremely flexible analog and digital resource routing
- Toggle register for output data
- Dedicated data input register (read-only)
- Interrupts
  - Two independent interrupt vectors
  - All GPIO pins are selectable as interrupts in EM0 and EM1
  - All EM4 wake-up pins are available as wakeups in EM0, EM1, and EM4
  - Separate enable, status, set and clear registers
  - Asynchronous sensing
  - Rising, falling or both edges
- Peripheral Reflex System producer
  - All GPIO pins are selectable

25.3 Functional Description

An overview of the GPIO module is shown in Figure 25.1. Pin Configuration. The GPIO pins are grouped into 16-pin ports. Each individual GPIO pin is called Pxn where x indicates the port (A, B, C ...) and n indicates the pin number (0,1,....,15). Fewer than 16 pins may be available on some ports depending on the total number of I/O pins on the package. After a reset, both input and output are disabled for all pins on the device, except for the Serial Wire Debug pins.

To use a pin, the Mode Register (GPIO_PORTx_MODEL/GPIO_PORTx_MODEH) must be configured for the pin to make it an input or output. These registers can also do more advanced configuration, which is covered in 25.3.1 Pin Configuration. When the port is configured as an input or an output, the Data In Register (GPIO_PORTx_DIN) can be used to read the level of each pin in the port (bit n in the register is connected to pin n on the port). When configured as an output, the value of the Data Out Register (GPIO_PORTx_DOUT) will be driven to the pin.

The DOUT value can be changed in the following ways:

- Writing to the GPIO_PORTx_DOUT register
- Writing the SET address of the GPIO_PORTx_DOUT register sets the corresponding DOUT bits
- Writing the CLEAR address of the GPIO_PORTx_DOUT register clears the corresponding DOUT bits
- Writing the TGL address of the GPIO_PORTx_DOUT register toggles the corresponding DOUT bits

Reading the GPIO_PORTx_DOUT register will return its contents.

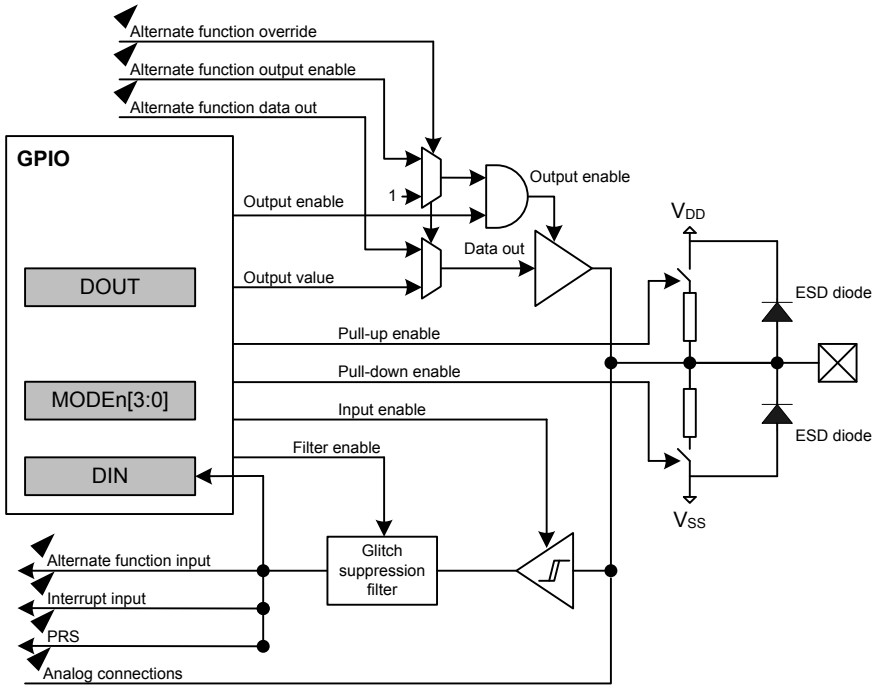


Figure 25.1. Pin Configuration

25.3.1 Pin Configuration

In addition to setting the pins as either outputs or inputs, the GPIO_PORTx_MODEL and GPIO_PORTx_MODEH registers can be used for more advanced configurations. GPIO_PORTx_MODEL contains 8 bit fields named MODEn (n=0,1,..7) which control pins 0-7, while GPIO_PORTx_MODEH contains 8 bit fields named MODEn (n=8,9,..15) which control pins 8-15. In some modes GPIO_PORTx_DOUT is also used for extra configurations like pull-up/down and glitch suppression filter enable. Table 25.1. Pin Configuration shows the available configurations.

Table 25.1. Pin Configuration

MODEn	Input	Output	DOUT	Pull-down	Pull-up	Alt Port Ctrl	Input Filter	Description
DISABLED	Disabled	Disabled	0					Input disabled
			1		On			Input disabled with pull-up
INPUT	Enabled if not DINDIS		0					Input enabled
			1				On	Input enabled with filter
INPUTPULL			0	On				Input enabled with pull-down
			1		On			Input enabled with pull-up
INPUTPULLFILTER			0	On			On	Input enabled with pull-down and filter
			1		On		On	Input enabled with pull-up and filter
PUSHPULL		Push-pull	x					Push-pull
PUSHPULLALT			x			On		Push-pull with alternate port control values
WIREDOR		Open Source (Wired-OR)	x					Open-source
WIREDORPULLDOWN			x	On				Open-source with pull-down
WIREDAND		Open Drain (Wired-AND)	x					Open-drain
WIREDANDFILTER			x				On	Open-drain with filter
WIREDANDPULLUP			x		On			Open-drain with pull-up
WIREDANDPULLUPFILTER			x		On		On	Open-drain with pull-up and filter
WIREDANDALT			x			On		Open-drain with alternate port control values
WIREDANDALTFILTER			x			On	On	Open-drain with alternate port control values and filter
WIREDANDALTPULLUP			x		On	On		Open-drain with alternate port control values and pull-up
WIREDANDALTPULLUPFILTER			x		On	On	On	Open-drain with alternate port control values, pull-up and filter

MODEn determines which mode the pin is in at a given time. Setting MODEn to DISABLED disables the pin, reducing power consumption to a minimum. When the output driver and input driver are disabled, the pin can be used as a connection for an analog module. An input is enabled by setting MODEn to any value other than DISABLED while DINDIS for the given port is cleared. Set DINDIS to disable the input of a GPIO port. The pull-up, pull-down and glitch filter function can optionally be applied to the input, see Figure 25.2. Tristated Output with Optional Pull-up or Pull-down.

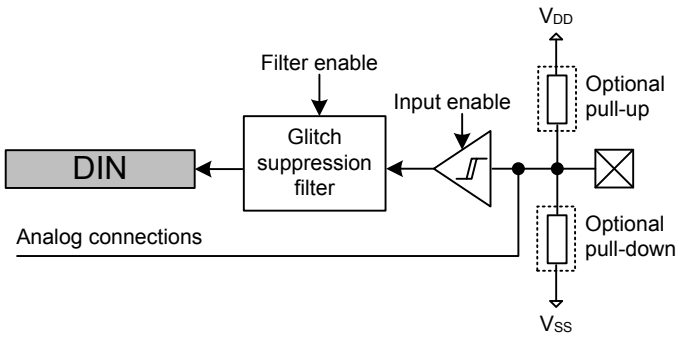


Figure 25.2. Tristated Output with Optional Pull-up or Pull-down

When MODEn is PUSHPULL or PUSHPULLALT, the pin operates in push-pull mode. In this mode, the pin can have alternate port control values and can be driven either high or low, dependent on the value of GPIO_PORTx_DOUT. The push-pull configuration is shown in Figure 25.3. Push-Pull Configuration.

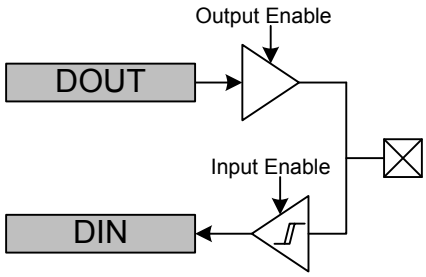


Figure 25.3. Push-Pull Configuration

When MODEn is WIREDOR or WIREDORPULLDOWN, the pin operates in open-source mode (with a pull-down resistor for WIREDORPULLDOWN). When driving a high value in open-source mode, the pull-down is disconnected to save power.

When the mode is prefixed with WIREDAND, the pin operates in open-drain mode as shown in Figure 25.4. Open-drain. In open-drain mode, the pin can have an input filter, a pull-up, alternate port control values or any combination of these. When driving a low value in open-drain mode, the pull-up is disconnected to save power.

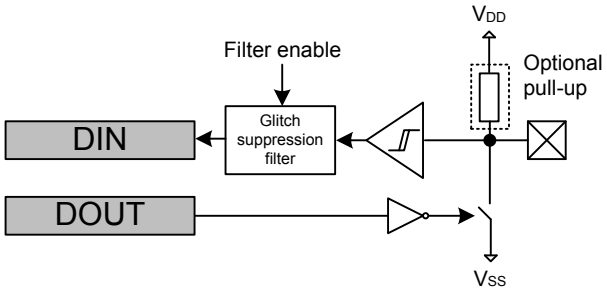


Figure 25.4. Open-drain

25.3.2 Low Noise Mode

When the ABUS is used to route GPIO to analog blocks, a low noise mode option is available to reduce the input noise. Low-noise mode is enabled on a pin-by-pin basis by setting bits in the GPIO_PORTx_AMUXMODE.LNEN bitfield to '1'. For example, GPIO_PORTA_AMUXMODE.LNEN[0] controls this mode for PA00.

By default, the bits of GPIO_PORTx_AMUXMODE.LNEN are reset to '0' and all pins will be in low leakage mode, which may be desired for inputs with high series impedance. When GPIO pins are configured for digital functions, they will always use low leakage mode, and setting the bit in GPIO_PORTx_AMUXMODE.LNEN has no effect.

25.3.3 Alternate Port Control

The Alternate Port Control allows for additional flexibility of port level settings. A user may setup two different port configurations (normal and alternate modes) and select which is applied on a pin by pin bases. For example you may configure half of port A to use the slowest slew rate while the other half uses a faster slew rate.

Alternate port control is enabled when MODEn is set to any of the ALT enumerated modes (i.e.. PUSHPULLALT). When MODEn is an alternate mode, the pin uses the alternate port control values specified in the DINDISALT and SLEWRATEALT fields in GPIO_PORTx_CTRL. In all other modes, the port control values are used from the DINDIS and SLEWRATE fields in GPIO_PORTx_CTRL.

25.3.4 Slew Rate

The slewrate can be applied to pins on a port-by-port basis. The slew rate applied to pins configured using normal MODEn settings can be controlled using the SLEWRATE fields in GPIO_PORTx_CTRL. The slewrate applied to pins configured using the alternate MODEn settings can be controlled using the SLEWRATEALT field.

The lowest slew rate setting has limited drive strength. That is the current is limited to about 1 mA. This setting provides slow switching and limited drive. A slew rate setting of 1 provides the slowest switching with full drive capability. The maximum recommended setting for most digital I/O is 6. A slew rate setting of 7 should only be used for high-speed clock signals, above 10 MHz. A setting of 7 should not be used on more than one pin per port. Please refer to the data sheet for GPIO rise and fall times.

25.3.5 Input Disable

The pin inputs can be disabled on a port-by-port basis. The input of pins configured using the normal MODEn settings can be disabled by setting DINDIS in GPIO_PORTx_CTRL. The input of pins configured using the alternate MODEn settings can be disabled by setting DINDISALT.

25.3.6 Configuration Lock

The GPIO configuration registers (GPIO_PORTx_CTRL, GPIO_PORTx_MODEL/H, GPIO_PORTx_AMUXMODE, GPIO_xBUSALLOC, GPIO_EXTIPSELL/H, GPIO_EXTIPINSELL/H, GPIO_x_yROUTE, and GPIO_xROUTEEN) can be locked by writing any value other than 0xA534 to GPIO_LOCK. Writing the value 0xA534 to the GPIO_LOCK register unlocks the configuration registers.

25.3.7 EM4 Functionality

By default GPIO pins revert back to their reset state when EM4 is entered. The GPIO pins can be configured to retain the settings for output enable, output value, pull enable, and pull direction while in EM4.

EM4 GPIO retention is controlled with the EM4IORETMODE field in the EMU_EM4CTRL register:

- Setting EM4IORETMODE to EM4EXIT will cause GPIO retention to persist while in EM4. GPIO state will be reset during wakeup.
- Setting EM4IORETMODE to SWUNLATCH will cause the GPIO retention to persist through EM4 and wakeup, until the EM4UNLATCH bit is written by software. When using SWUNLATCH, the GPIO register values are still reset on wakeup. To ensure the GPIO state does not change, software must re-write the GPIO registers before setting EM4UNLATCH and ending EM4 GPIO retention. Note that the GPIO state cannot be retained through an EM4 wakeup due to a reset (e.g., pin reset or POR reset) - only non-reset methods of EM4 wakeup are supported (e.g., EM4WU IRQ or BURTC IRQ).

See the EMU chapter for additional documentation on EM4IORETMODE and the EM4UNLATCH bit.

25.3.8 EM4 Wakeup

It is possible to trigger a wake-up from EM4 using any of the selectable EM4WU GPIO pins. The wake-up request can be triggered through the pins by enabling the corresponding bit in the GPIO_EM4WUEN register. When EM4 wake-up is enabled for the pin, the input filter is enabled during EM4. This is done to avoid false wake-up caused by glitches. In addition, the polarity of the EM4 wake-up request can be selected using the GPIO_EM4WUPOL register.

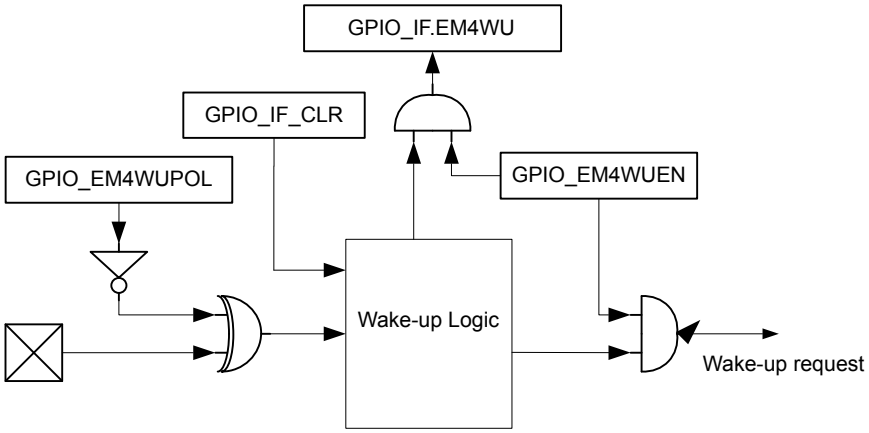


Figure 25.5. EM4 Wake-up Logic

The pins used for EM4 wake-up must be configured as inputs with glitch filters using the GPIO_PORTx_MODEL/H registers. If the input is disabled and the wakeup polarity is low, a false wakeup will occur when entering EM4. If the input is enabled, the glitch filtered is disabled, and the polarity is set low, a glitch will occur when going into EM4 that will cause an immediate wake-up. Before going down to EM4, it is important to clear the wake-up logic by setting the GPIO_IF_CLR bits, which clear the wake-up logic, including the GPIO_IF register. It is possible to determine which pin caused the EM4WU by reading the GPIO_IF register.

Each EM4WU signal is connected to a fixed pin. Refer to the Alternate Function Table for the location of each EM4 wakeup signal.

25.3.9 Debug Connections

25.3.9.1 JTAG Debug Connection

The JTAG Debug Port is a fixed location resource connected directly to specific GPIO pins.

Table 25.2. JTAG Debug Signals

JTAG Signal	GPIO Routing
TCK	GPIO.SWCLK
TMS	GPIO.SWDIO
TDO	GPIO.TDO
TDI	GPIO.TDI

Refer to the Alternate Function Table for the location of the JTAG signals. By default TMS, TCK, TDO, and TDI pin connections are enabled with internal pull up, pull down, no pull, and pull up resistors, respectively. It is possible to disable these pin connections (and disable the pull resistors) by setting the SWDIOTMSPEN, SWCLKTCKPEN, TDOPEN, and TDIPEN bits in GPIO_DBGROUTEPEN to 0.

25.3.9.2 Serial Wire Debug Connection

The SW Debug Port is a fixed location resource connected directly to specific GPIO pins.

Table 25.3. Serial Wire Debug Signals

Serial Wire Signal	GPIO Routing
SWCLK	GPIO.SWCLK
SWDIO	GPIO.SWDIO
SWV	GPIO.SWV

Refer to the Alternate Function Table for the location of the SW Debug port signals. The SWDIO and SWCLK pin connections are enabled by default with internal pull up and pull down resistors, respectively. It is possible to disable these pin connections (and disable the pull resistors) by setting the SWDIOTMSPEN and SWCLKTCKPEN bits in GPIO_DBGROUTEPEN to 0.

The Serial Wire Viewer pin, SWV, can be enabled by setting the SWVPEN bit in GPIO_TRACEROUTEPEN.

Note: The SWV pin is not affected by debug lock, so the SWV pin should not be enabled for production devices.

25.3.9.3 Disabling Debug Connections

When the debug pins are disabled, the device can no longer be accessed by a debugger. A reset will set the debug pins back to their enabled default state. The GPIO_DBGROUTEPEN register can only be updated when the debugger is disconnected from the system. Any attempts to modify GPIO_DBGROUTEPEN when the debugger is connected will not occur. If you do disable the debug pins, make sure you have at least a 3 second timeout at the start of your program code before you disable the debug pins. This way the debugger will have time to connect to the device after a reset and before the pins are disabled.

25.3.9.4 ETM Trace Connections

The device includes ETM trace pins. The trace clock can be enabled by setting the TRACECLKPEN bit-field in GPIO_TRACEROUTEPEN. The data pin(s) can be enabled individually by setting TRACEDATAxPEN in GPIO_TRACEROUTEPEN. The trace pins are fixed location resources connected to specific pins. Refer to the Alternate Function Table for the location of the ETM trace port signals.

25.3.10 Interrupt Generation

25.3.10.1 Standard Interrupt Generation

The GPIO can generate an interrupt from any edge of the input of any GPIO pin on the device. The standard interrupts have asynchronous sense capability, enabling wake-up from low energy modes. Ports may have restrictionson which energy mode(s) the standard interrupt is operable in. See Table 25.4. Standard Interrupt Energy Modes for details.

Table 25.4. Standard Interrupt Energy Modes

Ports	Supported Interrupt from Energy Modes
PA	EM0 and EM1
PB	EM0 and EM1
PC	EM0 and EM1
PD	EM0 and EM1

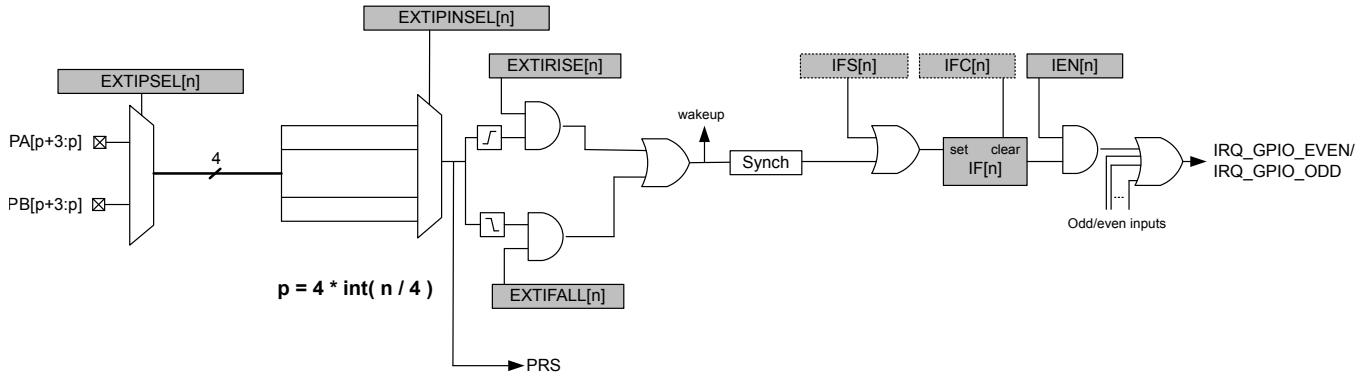


Figure 25.6. Pin n Interrupt Generation

The standard external pin interrupts are numbered starting with 0. Each interrupt has a corresponding enable bit in the GPIO_IEN register and an interrupt flag bit in the GPIO_IF register. Each interrupt may be used with one of four possible pins on any available port. First select the desired port for each interrupt using the corresponding EXTIPSELx field in the GPIO_EXTIPSELL register; (Some devices with many pins may also have a GPIO_EXTIPSELH register.)

Each interrupt can be mapped to one of four possible pins on the selected port. External interrupts EXTI0 through EXTI3 may be mapped to pins 0,1,2, or 3 on any available port. External interrupts EXTI4 through EXTI7 may be mapped to pins 4,5,6 or 7 on any available port.

Note: Note that while the EXTIEN field in the GPIO_IEN register has 15 bits, the number of useful bits is limited by the number of pins available in the widest port. If the widest port is 8 bits wide, only the first 8 external interrupts are useful.

The selected pin for each interrupt is the base plus the offset. The base for EXTI0 through EXTI3 is 0, while the base for interrupts EXTI4 through EXTI7 is 4. The base may be calculated by taking the interrupt number, dividing by four, then using only the integer portion of the quotient. (BASE = Integer(N/4)

The offset is selected using the corresponding field in the GPIO_EXTIPINSELL register, (Some devices with many pins may also have a GPIO_EXTIPINSELH register.) Subtract the base from the desired pin number to get the offset. For example, to map EXTI5 to pin 7 of PORTA, the base is 4 and the offset will be 3.

The GPIO_EXTIRISE[n] and GPIO_EXTIFALL[n] registers enable sensing of rising and falling edges. By setting the EXT[n] bit in GPIO_IEN, a high interrupt flag n, will trigger one of two interrupt lines. The even interrupt line is triggered by any enabled even numbered interrupt flag index, while the odd interrupt line is triggered by odd flag indexes. The interrupt flags can be set and cleared by software when writing the GPIO_IF_SET and GPIO_IF_CLR register locations. Since the external interrupts are asynchronous, they are sensitive to noise. To increase noise tolerance, the MODEx field(s) in the GPIO_PORTx_MODEL/H registers should be set to include glitch filtering for pins that have external interrupts enabled.

25.3.10.2 Interrupt Generation on EM4WU Pins

EM4WU (EM4 wake-up) signals on PA, PB, PC or PD may be used to generate edge-sensitive interrupts in EM0, EM1, and EM4.

In order to enable an EM4WU pin as an interrupt, set the EM4WUIENn field in the GPIO_IEN register and the EM4WUENn field in the EM4WUEN register. The EM4WUPOLn field in the GPIO_EM4WUPOL register is used to set the desired polarity for the interrupt (0 for a falling edge, and 1 for a rising edge).

Upon an interrupt occurring, the corresponding EM4WU index in the GPIO_IF register will be set along with the odd or even interrupt line depending on the index inside of GPIO_IF. For example, by setting EM4WU8 in GPIO_EM4WUPOL and EM4WU[8] in GPIO_IEN, the interrupt flag EM4WU[8] in GPIO_IF will be triggered by a rising edge on pin EM4WU8 and a interrupt request will be sent on the GPIO_EVEN IRQ.

The wake-up granularity of the EM4WU interrupts is based on the settings of the EM4WU field in the GPIO_IEN register and the EM4WUEN field in the GPIO_EM4WUEN register (see Table 25.5. EM4WU Interrupt Energy Mode Wakeup).

Table 25.5. EM4WU Interrupt Energy Mode Wakeup

EM4WUIENn in GPIO_IEN	EM4WUENn in GPIO_EM4WUEN	Energy Mode Wakeup	Interrupt
x	0	No Wake	No Interrupt
0	1	Wake from EM4	No Interrupt
1	1	Wake from EM0, EM1, and EM4	Interrupt from EM0 and EM1

For example, to configure the device to wake up and generate an interrupt when PD02 (EM4WU9) sees a falling edge:

1. Set bit 9 of EM4WUEN in the GPIO_EM4WUEN register to '1'. This enables the asynchronous wake logic.
2. Set bit 9 of EM4WUIEN in the GPIO_IEN register to '1'. This enables routing of the wake signal to the GPIO_ODD IRQ.
3. Clear bit 9 of EM4WUPOL in the GPIO_EM4WUPOL register to '0'. This indicates that the interrupt should occur when a falling edge is detected at the pin.
4. Enable the GPIO_ODD IRQ. The ODD interrupt is used because the bit index of EM4WUIF in GPIO_IF is odd.

25.3.11 Output to PRS

All pins within a group of four(0-3,4-7,8-11,12-15) from all ports are grouped together to form one PRS producer which outputs to the PRS. The pin from which the output should be taken is selected in the same fashion as the edge interrupts.

PRS output is not affected by the interrupt edge detection logic or gated by the IEN bits. See 25.3.10 Interrupt Generation for an illustration of where the PRS output signal is generated.

25.3.12 Peripheral Resource Routing

Most peripherals have resources that need to be connected to GPIO pins to function. For example, the I2C has SDA and SCL which need to be connected to pins for the I2C to communicate with other ICs. Resources come in three types. Fixed resources are hard-wired to a pin and can only be accessed in that location. For example the LFXO.LFXTAL_I and LFXO.LFXTAL_O resources are only available on one pin each. Digital route-able resources are connected to pins through the 25.3.12.1 Digital Bus (DBUS) which allows for extremely flexible resource placement. Analog route-able resources are connected to pins though the 25.3.12.2 Analog Bus (ABUS) which provides extremely flexible resource placement.

25.3.12.1 Digital Bus (DBUS)

The Digital Bus (DBUS) is an any-to-any switch matrix between peripheral resources and GPIO pins as shown in Figure 25.7. Digital Bus Interconnect. There are two DBUSes on the SiXG301 - DBUSAB serves ports A and B, while DBUSCD serves ports C and D. Not all peripherals have access to both DBUSes.

To connect a resource to a pin, first select the desired PORT and PIN in the GPIO_x_yROUTE register, where x is the peripheral name and y is the resource name. The PORT field is encoded as PA = 0, PB = 1, PC = 2, etc. Once the pin is selected, the resource must be enabled by setting its enable bit in the appropriate GPIO_x_ROUTEEN register. For example, to route the SDA resource of I2C0 to PB03, set PORT to 0x1 and PIN to 0x3 in GPIO_I2C0_SDAROUTE. Then set the GPIO_I2C0_ROUTEEN.SDAPEN bit.

Any pin connected to a digital resource should be properly configured for that resource (refer to 25.3.1 Pin Configuration). For example, an I2C SDA should be configured as open-drain, a EUSART TX should be configured as push-pull, and a EUSART RX should be configured as an input.

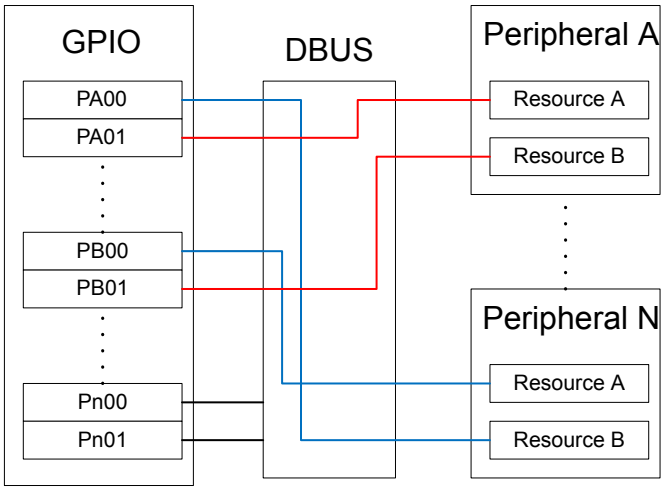


Figure 25.7. Digital Bus Interconnect

25.3.12.2 Analog Bus (ABUS)

Analog peripherals may be connected to any pins on port A, B, C, or D via the Analog Bus. There are three analog buses on the SiXG301: one dedicated to Port A (ABUSA), one dedicated to port B (ABUSB), and one that serves both ports C and D (ABUSCD). The specific pin and port selection for analog resources are configured in the analog peripherals. Refer to the respective analog peripheral chapter for this information. However, the GPIO block must be configured to grant the peripheral access to an ABUS before any connection can be made.

Note: The analog signals on ABUSes will be voltage limited by the lowest supply voltage of IOVDD and AVDD.

Up to two analog peripherals may be given access to an ABUS at any one time and the even/odd pins of each bus are configured independently. This means that a single bus may have up to four different analog peripherals connected to it: two on the even pins and two on the odd pins. The GPIO_xBUSALLOC register, where x is A, B, or CD, determines which peripherals have access to the bus. To grant a peripheral access to the bus even pins select it in either the EVEN0 or EVEN1 field. To grant a peripheral access to the bus odd pins select it in either the ODD0 or ODD1 fields.

When a differential connection is being used, positive inputs are restricted to the EVEN pins and negative inputs are restricted to the ODD pins. When a single ended connection is being used, the positive input is avaliable on all pins.

Peripherals may be given access to as many buses as desired. For example the ADC may be given access to ABUSA, ABUSB, and ABUSCD allowing it to select any pin on ports A-D. If two peripherals select the same port and pin the ABUS will make both connections simultaneously, effectively connecting the two peripherals together.

Any pin connected to an analog resource should be configured to input DISABLED as described in 25.3.1 Pin Configuration

The process for configuring an analog peripheral to access a pin through the ABUS is as follows:

- Configure the desired analog port pins to input DISABLED mode in the corresponding GPIO_PORTx_MODEL/H register.
- If needed, configure the desired GPIO for low noise or low leakage using the corresponding GPIO_PORTx_AMUXMODE.LNEN bit.
- Configure the corresponding GPIO_xBUSALLOC field to grant access to the desired peripheral on the desired ABUS.
- Configure the analog peripheral to select the desired port and channel as described in the peripheral chapter.

25.3.12.3 Pin Function Tables

This section details the functions and GPIO pins available on the most fully-featured devices in the SiXG301 family. Availability of GPIO and signals varies. Refer to the device data sheet for specific peripheral and GPIO availability. Fixed-pin peripheral resources are shown in Table 25.6. GPIO Alternate Function Table, ABUS routing options are listed in Table 25.7. ABUS Routing Table, and DBUS routing options are listed in Table 25.8. DBUS Routing Table

Table 25.6. GPIO Alternate Function Table

GPIO	Alternate Functions	QFN40 Max GPIO Package	QFN40 with External Flash Package	QFN32 Max GPIO Package	QFN32 with LED Pre-Drive Package
PA00	ADC0.VREFP	Yes	Yes	Yes	Yes
PA01	GPIO.SWCLK	Yes	Yes	Yes	Yes
PA02	GPIO.SWDIO	Yes	Yes	Yes	Yes
PA03	GPIO.SWV	Yes		Yes	Yes
	GPIO.TDO	Yes		Yes	Yes
	GPIO.TRACEDATA0	Yes		Yes	Yes
PA04	GPIO.TDI	Yes		Yes	Yes
	GPIO.TRACECLK	Yes		Yes	Yes
PA05	GPIO.TRACEDATA1	Yes			
	GPIO.EM4WU0	Yes			
PA06	GPIO.TRACEDATA2	Yes			
PA07	GPIO.TRACEDATA3	Yes			
PB01	ETAMPDET.ETAMPIN0	Yes	Yes	Yes	Yes
	GPIO.EM4WU3	Yes	Yes	Yes	Yes
PB03	GPIO.EM4WU4	Yes	Yes		
PC00	ETAMPDET.ETAMPIN1	Yes	Yes	Yes	Yes
	GPIO.EM4WU6	Yes	Yes	Yes	Yes
PC01	ETAMPDET.ETAMPOUT0	Yes	Yes	Yes	Yes
PC02	ETAMPDET.ETAMPOUT1	Yes	Yes	Yes	Yes
PC05	GPIO.EM4WU7	Yes	Yes	Yes	Yes
PC07	GPIO.EM4WU8	Yes	Yes		
PD00	LFXO.LFXTAL_O	Yes	Yes	Yes	Yes
PD01	LFXO.LFXTAL_I	Yes	Yes	Yes	Yes
	LFXO.LF_EXTCLK	Yes	Yes	Yes	Yes
PD02	GPIO.EM4WU9	Yes	Yes	Yes	Yes
PD05	GPIO.EM4WU10	Yes	Yes	Yes	

Table 25.7. ABUS Routing Table

Peripheral	Signal	PA		PB		PC		PD	
		EVEN	ODD	EVEN	ODD	EVEN	ODD	EVEN	ODD
ACMP0	ana_neg	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	ana_pos	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ACMP1	ana_neg	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	ana_pos	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ADC0	ana_neg	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	ana_pos	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
LEDDRV0	drnsense					Yes	Yes	Yes	Yes
	ipksense					Yes	Yes	Yes	Yes
	vacsense					Yes	Yes	Yes	Yes

Table 25.8. DBUS Routing Table

Peripheral.Resource	PORT			
	PA	PB	PC	PD
ACMP0.DIGOUT	Available	Available	Available	Available
ACMP1.DIGOUT	Available	Available	Available	Available
CMU.CLKIN0			Available	Available
CMU.CLKOUT0			Available	Available
CMU.CLKOUT1			Available	Available
CMU.CLKOUT2	Available	Available		
EUSART0.CS	Available	Available		
EUSART0.CTS	Available	Available		
EUSART0.RTS	Available	Available		
EUSART0.RX	Available	Available		
EUSART0.SCLK	Available	Available		
EUSART0.TX	Available	Available		
EUSART1.CS	Available	Available	Available	Available
EUSART1.CTS	Available	Available	Available	Available
EUSART1.RTS	Available	Available	Available	Available
EUSART1.RX	Available	Available	Available	Available
EUSART1.SCLK	Available	Available	Available	Available
EUSART1.TX	Available	Available	Available	Available
EUSART2.CS			Available	Available
EUSART2.CTS			Available	Available
EUSART2.RTS			Available	Available
EUSART2.RX			Available	Available
EUSART2.SCLK			Available	Available
EUSART2.TX			Available	Available
FRC.DCLK			Available	Available
FRC.DFRAME			Available	Available
FRC.DOUT			Available	Available
I2C0.SCL	Available	Available	Available	Available
I2C0.SDA	Available	Available	Available	Available
I2C1.SCL			Available	Available
I2C1.SDA			Available	Available
I2C2.SCL			Available	Available
I2C2.SDA			Available	Available
LETIMER0.OUT0	Available	Available		
LETIMER0.OUT1	Available	Available		
LFXO.LFXO_CLK_LV_RAW			Available	Available
MODEM.ANT0	Available	Available	Available	Available
MODEM.ANT1	Available	Available	Available	Available

Peripheral.Resource	PORT			
	PA	PB	PC	PD
MODEM.ANT_ROLL_OVER			Available	Available
MODEM.ANT_RR0			Available	Available
MODEM.ANT_RR1			Available	Available
MODEM.ANT_RR2			Available	Available
MODEM.ANT_RR3			Available	Available
MODEM.ANT_RR4			Available	Available
MODEM.ANT_RR5			Available	Available
MODEM.ANT_SW_EN			Available	Available
MODEM.ANT_SW_US			Available	Available
MODEM.ANT_TRIG			Available	Available
MODEM.ANT_TRIG_STOP			Available	Available
MODEM.DCLK	Available	Available		
MODEM.DIN	Available	Available		
MODEM.DOUT	Available	Available		
PCNT0.S0IN	Available	Available		
PCNT0.S1IN	Available	Available		
PIXELRZ0.RZ_TX_OUT	Available	Available	Available	Available
PIXELRZ1.RZ_TX_OUT	Available	Available	Available	Available
PRS.ASYNCH0	Available	Available		
PRS.ASYNCH1	Available	Available		
PRS.ASYNCH10			Available	Available
PRS.ASYNCH11			Available	Available
PRS.ASYNCH2	Available	Available		
PRS.ASYNCH3	Available	Available		
PRS.ASYNCH4	Available	Available		
PRS.ASYNCH5	Available	Available		
PRS.ASYNCH6			Available	Available
PRS.ASYNCH7			Available	Available
PRS.ASYNCH8			Available	Available
PRS.ASYNCH9			Available	Available
PRS.SYNCH0	Available	Available	Available	Available
PRS.SYNCH1	Available	Available	Available	Available
PRS.SYNCH2	Available	Available	Available	Available
PRS.SYNCH3	Available	Available	Available	Available
RAC.LNAEN	Available	Available	Available	Available
RAC.PAEN	Available	Available	Available	Available
TIMER0.CC0	Available	Available	Available	Available
TIMER0.CC1	Available	Available	Available	Available
TIMER0.CC2	Available	Available	Available	Available
TIMER0.CDTI0	Available	Available	Available	Available

Peripheral.Resource	PORT			
	PA	PB	PC	PD
TIMER0.CDTI1	Available	Available	Available	Available
TIMER0.CDTI2	Available	Available	Available	Available
TIMER1.CC0	Available	Available	Available	Available
TIMER1.CC1	Available	Available	Available	Available
TIMER1.CC2	Available	Available	Available	Available
TIMER1.CDTI0	Available	Available	Available	Available
TIMER1.CDTI1	Available	Available	Available	Available
TIMER1.CDTI2	Available	Available	Available	Available
TIMER2.CC0	Available	Available	Available	Available
TIMER2.CC1	Available	Available	Available	Available
TIMER2.CC2	Available	Available	Available	Available
TIMER2.CC3	Available	Available	Available	Available
TIMER2.CC4	Available	Available	Available	Available
TIMER2.CC5	Available	Available	Available	Available
TIMER2.CC6	Available	Available	Available	Available
TIMER2.CDTI0	Available	Available	Available	Available
TIMER2.CDTI1	Available	Available	Available	Available
TIMER2.CDTI2	Available	Available	Available	Available
TIMER3.CC0			Available	Available
TIMER3.CC1			Available	Available
TIMER3.CC2			Available	Available
TIMER3.CC3			Available	Available
TIMER3.CC4			Available	Available
TIMER3.CC5			Available	Available
TIMER3.CC6			Available	Available
TIMER3.CDTI0			Available	Available
TIMER3.CDTI1			Available	Available
TIMER3.CDTI2			Available	Available

25.4 Synchronization

To avoid metastability in synchronous logic connected to the pins, all inputs are synchronized with double flip-flops. The flip-flops for the input data run on the selected APB clock for the GPIO module (PCLK). Consequently, when a pin changes state, the change will propagate to GPIO_PORTx_DIN after two 2 PCLK cycles. Synchronization (also running on the PCLK) is also added for interrupt input. To save power when the external interrupts are not used, the synchronization flip-flops for these can be turned off by clearing the EXTINT field in the GPIO_IEN register.

25.5 GPIO Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	GPIO_IPVERSION	R	Main
0x030	GPIO_PORTA_CTRL	RW	Port Control
0x034	GPIO_PORTA_MODEL	RW	Mode Low
0x03C	GPIO_PORTA_MODEH	RW	Mode High
0x040	GPIO_PORTA_DOUT	RW	Data Out
0x044	GPIO_PORTA_DIN	RH	Data in
0x048	GPIO_PORTA_AMUXMODE	RW	AMUXMODE
0x060	GPIO_PORTB_CTRL	RW	Port Control
0x064	GPIO_PORTB_MODEL	RW	Mode Low
0x070	GPIO_PORTB_DOUT	RW	Data Out
0x074	GPIO_PORTB_DIN	RH	Data in
0x078	GPIO_PORTB_AMUXMODE	RW	AMUXMODE
0x090	GPIO_PORTC_CTRL	RW	Port Control
0x094	GPIO_PORTC_MODEL	RW	Mode Low
0x0A0	GPIO_PORTC_DOUT	RW	Data Out
0x0A4	GPIO_PORTC_DIN	RH	Data in
0x0A8	GPIO_PORTC_AMUXMODE	RW	AMUXMODE
0x0C0	GPIO_PORTD_CTRL	RW	Port Control
0x0C4	GPIO_PORTD_MODEL	RW	Mode Low
0x0D0	GPIO_PORTD_DOUT	RW	Data Out
0x0D4	GPIO_PORTD_DIN	RH	Data in
0x0D8	GPIO_PORTD_AMUXMODE	RW	AMUXMODE
0x300	GPIO_LOCK	W	Lock Register
0x310	GPIO_GPIOLOCKSTATUS	RH	Lock Status
0x320	GPIO_ABUSALLOC	RW	A Bus Allocation
0x324	GPIO_BBUSALLOC	RW	B Bus Allocation
0x328	GPIO_CDBUSALLOC	RW	CD Bus Allocation
0x400	GPIO_EXTIPSELL	RW	External Interrupt Port Select Low
0x404	GPIO_EXTIPSELH	RW	External Interrupt Port Select High
0x408	GPIO_EXTIPINSELL	RW	External Interrupt Pin Select Low
0x40C	GPIO_EXTIPINSELH	RW	External Interrupt Pin Select High
0x410	GPIO_EXTIRISE	RW	External Interrupt Rising Edge Trigger
0x414	GPIO_EXTIFALL	RW	External Interrupt Falling Edge Trigger
0x420	GPIO_IF	RWH INTFLAG	Interrupt Flag
0x424	GPIO_IEN	RW	Interrupt Enable
0x42C	GPIO_EM4WUEN	RW	EM4 Wakeup Enable
0x430	GPIO_EM4WUPOL	RW	EM4 Wakeup Polarity
0x440	GPIO_DBGROUTEPEN	RW	Debugger Route Pin Enable
0x444	GPIO_TRACEROUTEPEN	RW	Trace Route Pin Enable

Offset	Name	Type	Description
0x450	GPIO_ACMP0_ROUTEEN	RW	ACMP0 Pin Enable
0x454	GPIO_ACMP0_ACMPOUT-ROUTE	RW	ACMPOUT Port/Pin Select
0x45C	GPIO_ACMP1_ROUTEEN	RW	ACMP1 Pin Enable
0x460	GPIO_ACMP1_ACMPOUT-ROUTE	RW	ACMPOUT Port/Pin Select
0x468	GPIO_EUSART0_ROUTEEN	RW	EUSART0 Pin Enable
0x46C	GPIO_EUSART0_CSROUTE	RW	CS Port/Pin Select
0x470	GPIO_EUSART0_CTSROUTE	RW	CTS Port/Pin Select
0x474	GPIO_EUSART0_RTSROUTE	RW	RTS Port/Pin Select
0x478	GPIO_EUSART0_RXROUTE	RW	RX Port/Pin Select
0x47C	GPIO_EUSART0_SCLKROUTE	RW	SCLK Port/Pin Select
0x480	GPIO_EUSART0_TXROUTE	RW	TX Port/Pin Select
0x488	GPIO_EUSART1_ROUTEEN	RW	EUSART1 Pin Enable
0x48C	GPIO_EUSART1_CSROUTE	RW	CS Port/Pin Select
0x490	GPIO_EUSART1_CTSROUTE	RW	CTS Port/Pin Select
0x494	GPIO_EUSART1_RTSROUTE	RW	RTS Port/Pin Select
0x498	GPIO_EUSART1_RXROUTE	RW	RX Port/Pin Select
0x49C	GPIO_EUSART1_SCLKROUTE	RW	SCLK Port/Pin Select
0x4A0	GPIO_EUSART1_TXROUTE	RW	TX Port/Pin Select
0x4A8	GPIO_EUSART2_ROUTEEN	RW	EUSART2 Pin Enable
0x4AC	GPIO_EUSART2_CSROUTE	RW	CS Port/Pin Select
0x4B0	GPIO_EUSART2_CTSROUTE	RW	CTS Port/Pin Select
0x4B4	GPIO_EUSART2_RTSROUTE	RW	RTS Port/Pin Select
0x4B8	GPIO_EUSART2_RXROUTE	RW	RX Port/Pin Select
0x4BC	GPIO_EUSART2_SCLKROUTE	RW	SCLK Port/Pin Select
0x4C0	GPIO_EUSART2_TXROUTE	RW	TX Port/Pin Select
0x4C8	GPIO_FRC_ROUTEEN	RW	FRC Pin Enable
0x4CC	GPIO_FRC_DCLKROUTE	RW	DCLK Port/Pin Select
0x4D0	GPIO_FRC_DFRAMEROUTE	RW	DFRAME Port/Pin Select
0x4D4	GPIO_FRC_DOUTROUTE	RW	DOUT Port/Pin Select
0x4DC	GPIO_CMU_ROUTEEN	RW	HOSTCMU Pin Enable
0x4E0	GPIO_CMU_CLKIN0ROUTE	RW	CLKIN0 Port/Pin Select
0x4E4	GPIO_CMU_CLKOUT0ROUTE	RW	CLKOUT0 Port/Pin Select
0x4E8	GPIO_CMU_CLKOUT1ROUTE	RW	CLKOUT1 Port/Pin Select
0x4EC	GPIO_CMU_CLKOUT2ROUTE	RW	CLKOUT2 Port/Pin Select
0x4F8	GPIO_I2C0_ROUTEEN	RW	I2C0 Pin Enable
0x4FC	GPIO_I2C0_SCLROUTE	RW	SCL Port/Pin Select
0x500	GPIO_I2C0_SDAROUTE	RW	SDA Port/Pin Select
0x508	GPIO_I2C1_ROUTEEN	RW	I2C1 Pin Enable
0x50C	GPIO_I2C1_SCLROUTE	RW	SCL Port/Pin Select
0x510	GPIO_I2C1_SDAROUTE	RW	SDA Port/Pin Select

Offset	Name	Type	Description
0x518	GPIO_I2C2_ROUTEEN	RW	I2C2 Pin Enable
0x51C	GPIO_I2C2_SCLROUTE	RW	SCL Port/Pin Select
0x520	GPIO_I2C2_SDAROUTE	RW	SDA Port/Pin Select
0x528	GPIO_LETIMER0_ROUTEEN	RW	LETIMER0 Pin Enable
0x52C	GPIO_LETIMER0_OUT0ROUTE	RW	OUT0 Port/Pin Select
0x530	GPIO_LETIMER0_OUT1ROUTE	RW	OUT1 Port/Pin Select
0x538	GPIO_LFXO_ROUTEEN	RW	LFXO Pin Enable
0x53C	GPIO_LFXO_LFXOCLKLVRAW-ROUTE	RW	LFXOCLKLVRAW Port/Pin Select
0x544	GPIO_MODEM_ROUTEEN	RW	MODEM Pin Enable
0x548	GPIO_MODEM_ANT0ROUTE	RW	ANT0 Port/Pin Select
0x54C	GPIO_MODEM_ANT1ROUTE	RW	ANT1 Port/Pin Select
0x550	GPIO_MODEM_ANTROLLO-VERROUTE	RW	ANTROLLOVER Port/Pin Select
0x554	GPIO_MO-DEM_ANTRR0ROUTE	RW	ANTRR0 Port/Pin Select
0x558	GPIO_MO-DEM_ANTRR1ROUTE	RW	ANTRR1 Port/Pin Select
0x55C	GPIO_MO-DEM_ANTRR2ROUTE	RW	ANTRR2 Port/Pin Select
0x560	GPIO_MO-DEM_ANTRR3ROUTE	RW	ANTRR3 Port/Pin Select
0x564	GPIO_MO-DEM_ANTRR4ROUTE	RW	ANTRR4 Port/Pin Select
0x568	GPIO_MO-DEM_ANTRR5ROUTE	RW	ANTRR5 Port/Pin Select
0x56C	GPIO_MODEM_ANTSWEN-ROUTE	RW	ANTSWEN Port/Pin Select
0x570	GPIO_MODEM_ANTSWUS-ROUTE	RW	ANTSWUS Port/Pin Select
0x574	GPIO_MODEM_ANTTRIG-ROUTE	RW	ANTTRIG Port/Pin Select
0x578	GPIO_MODEM_ANTTRIGSTOP-ROUTE	RW	ANTTRIGSTOP Port/Pin Select
0x57C	GPIO_MODEM_DCLKROUTE	RW	DCLK Port/Pin Select
0x580	GPIO_MODEM_DINROUTE	RW	DIN Port/Pin Select
0x584	GPIO_MODEM_DOUTROUTE	RW	DOUT Port/Pin Select
0x590	GPIO_PCNT0_S0INROUTE	RW	S0IN Port/Pin Select
0x594	GPIO_PCNT0_S1INROUTE	RW	S1IN Port/Pin Select
0x59C	GPIO_PIXELRZ0_ROUTEEN	RW	PIXELRZ0 Pin Enable
0x5A0	GPIO_PIXELRZ0_RZTXOUT-ROUTE	RW	RZTXOUT Port/Pin Select
0x5A8	GPIO_PIXELRZ1_ROUTEEN	RW	PIXELRZ1 Pin Enable
0x5AC	GPIO_PIXELRZ1_RZTXOUT-ROUTE	RW	RZTXOUT Port/Pin Select
0x5B4	GPIO_PRS0_ROUTEEN	RW	PRS0 Pin Enable

Offset	Name	Type	Description
0x5B8	GPIO_PRS0_ASYNCH0ROUTE	RW	ASYNCH0 Port/Pin Select
0x5BC	GPIO_PRS0_ASYNCH1ROUTE	RW	ASYNCH1 Port/Pin Select
0x5C0	GPIO_PRS0_ASYNCH2ROUTE	RW	ASYNCH2 Port/Pin Select
0x5C4	GPIO_PRS0_ASYNCH3ROUTE	RW	ASYNCH3 Port/Pin Select
0x5C8	GPIO_PRS0_ASYNCH4ROUTE	RW	ASYNCH4 Port/Pin Select
0x5CC	GPIO_PRS0_ASYNCH5ROUTE	RW	ASYNCH5 Port/Pin Select
0x5D0	GPIO_PRS0_ASYNCH6ROUTE	RW	ASYNCH6 Port/Pin Select
0x5D4	GPIO_PRS0_ASYNCH7ROUTE	RW	ASYNCH7 Port/Pin Select
0x5D8	GPIO_PRS0_ASYNCH8ROUTE	RW	ASYNCH8 Port/Pin Select
0x5DC	GPIO_PRS0_ASYNCH9ROUTE	RW	ASYNCH9 Port/Pin Select
0x5E0	GPIO_PRS0_ASYNCH10ROUTE	RW	ASYNCH10 Port/Pin Select
0x5E4	GPIO_PRS0_ASYNCH11ROUTE	RW	ASYNCH11 Port/Pin Select
0x5E8	GPIO_PRS0_SYNCH0ROUTE	RW	SYNCH0 Port/Pin Select
0x5EC	GPIO_PRS0_SYNCH1ROUTE	RW	SYNCH1 Port/Pin Select
0x5F0	GPIO_PRS0_SYNCH2ROUTE	RW	SYNCH2 Port/Pin Select
0x5F4	GPIO_PRS0_SYNCH3ROUTE	RW	SYNCH3 Port/Pin Select
0x5FC	GPIO_RAC_ROUTEEN	RW	RAC Pin Enable
0x600	GPIO_RAC_LNAENROUTE	RW	LNAEN Port/Pin Select
0x604	GPIO_RAC_PAENROUTE	RW	PAEN Port/Pin Select
0x668	GPIO_TIMER0_ROUTEEN	RW	TIMER0 Pin Enable
0x66C	GPIO_TIMER0_CC0ROUTE	RW	CC0 Port/Pin Select
0x670	GPIO_TIMER0_CC1ROUTE	RW	CC1 Port/Pin Select
0x674	GPIO_TIMER0_CC2ROUTE	RW	CC2 Port/Pin Select
0x678	GPIO_TIMER0_CDTI0ROUTE	RW	CDTI0 Port/Pin Select
0x67C	GPIO_TIMER0_CDTI1ROUTE	RW	CDTI1 Port/Pin Select
0x680	GPIO_TIMER0_CDTI2ROUTE	RW	CDTI2 Port/Pin Select
0x688	GPIO_TIMER1_ROUTEEN	RW	TIMER1 Pin Enable
0x68C	GPIO_TIMER1_CC0ROUTE	RW	CC0 Port/Pin Select
0x690	GPIO_TIMER1_CC1ROUTE	RW	CC1 Port/Pin Select
0x694	GPIO_TIMER1_CC2ROUTE	RW	CC2 Port/Pin Select
0x698	GPIO_TIMER1_CDTI0ROUTE	RW	CDTI0 Port/Pin Select
0x69C	GPIO_TIMER1_CDTI1ROUTE	RW	CDTI1 Port/Pin Select
0x6A0	GPIO_TIMER1_CDTI2ROUTE	RW	CDTI2 Port/Pin Select
0x6A8	GPIO_TIMER2_ROUTEEN	RW	TIMER2 Pin Enable
0x6AC	GPIO_TIMER2_CC0ROUTE	RW	CC0 Port/Pin Select
0x6B0	GPIO_TIMER2_CC1ROUTE	RW	CC1 Port/Pin Select
0x6B4	GPIO_TIMER2_CC2ROUTE	RW	CC2 Port/Pin Select
0x6B8	GPIO_TIMER2_CC3ROUTE	RW	CC3 Port/Pin Select
0x6BC	GPIO_TIMER2_CC4ROUTE	RW	CC4 Port/Pin Select
0x6C0	GPIO_TIMER2_CC5ROUTE	RW	CC5 Port/Pin Select

Offset	Name	Type	Description
0x6C4	GPIO_TIMER2_CC6ROUTE	RW	CC6 Port/Pin Select
0x6C8	GPIO_TIMER2_CDTI0ROUTE	RW	CDTI0 Port/Pin Select
0x6CC	GPIO_TIMER2_CDTI1ROUTE	RW	CDTI1 Port/Pin Select
0x6D0	GPIO_TIMER2_CDTI2ROUTE	RW	CDTI2 Port/Pin Select
0x6D4	GPIO_TIMER2_CDTI3ROUTE	RW	CDTI3 Port/Pin Select
0x6D8	GPIO_TIMER2_CDTI4ROUTE	RW	CDTI4 Port/Pin Select
0x6DC	GPIO_TIMER2_CDTI5ROUTE	RW	CDTI5 Port/Pin Select
0x6E0	GPIO_TIMER2_CDTI6ROUTE	RW	CDTI6 Port/Pin Select
0x6E8	GPIO_TIMER3_ROUTEEN	RW	TIMER3 Pin Enable
0x6EC	GPIO_TIMER3_CC0ROUTE	RW	CC0 Port/Pin Select
0x6F0	GPIO_TIMER3_CC1ROUTE	RW	CC1 Port/Pin Select
0x6F4	GPIO_TIMER3_CC2ROUTE	RW	CC2 Port/Pin Select
0x6F8	GPIO_TIMER3_CC3ROUTE	RW	CC3 Port/Pin Select
0x6FC	GPIO_TIMER3_CC4ROUTE	RW	CC4 Port/Pin Select
0x700	GPIO_TIMER3_CC5ROUTE	RW	CC5 Port/Pin Select
0x704	GPIO_TIMER3_CC6ROUTE	RW	CC6 Port/Pin Select
0x708	GPIO_TIMER3_CDTI0ROUTE	RW	CDTI0 Port/Pin Select
0x70C	GPIO_TIMER3_CDTI1ROUTE	RW	CDTI1 Port/Pin Select
0x710	GPIO_TIMER3_CDTI2ROUTE	RW	CDTI2 Port/Pin Select
0x714	GPIO_TIMER3_CDTI3ROUTE	RW	CDTI3 Port/Pin Select
0x718	GPIO_TIMER3_CDTI4ROUTE	RW	CDTI4 Port/Pin Select
0x71C	GPIO_TIMER3_CDTI5ROUTE	RW	CDTI5 Port/Pin Select
0x720	GPIO_TIMER3_CDTI6ROUTE	RW	CDTI6 Port/Pin Select
0x1000	GPIO_IPVERSION_SET	R	Main
0x1030	GPIO_PORTA_CTRL_SET	RW	Port Control
0x1034	GPIO_PORTA_MODEL_SET	RW	Mode Low
0x103C	GPIO_PORTA_MODEH_SET	RW	Mode High
0x1040	GPIO_PORTA_DOUT_SET	RW	Data Out
0x1044	GPIO_PORTA_DIN_SET	RH	Data in
0x1048	GPIO_PORTA_AMUX-MODE_SET	RW	AMUXMODE
0x1060	GPIO_PORTB_CTRL_SET	RW	Port Control
0x1064	GPIO_PORTB_MODEL_SET	RW	Mode Low
0x1070	GPIO_PORTB_DOUT_SET	RW	Data Out
0x1074	GPIO_PORTB_DIN_SET	RH	Data in
0x1078	GPIO_PORTB_AMUX-MODE_SET	RW	AMUXMODE
0x1090	GPIO_PORTC_CTRL_SET	RW	Port Control
0x1094	GPIO_PORTC_MODEL_SET	RW	Mode Low
0x10A0	GPIO_PORTC_DOUT_SET	RW	Data Out
0x10A4	GPIO_PORTC_DIN_SET	RH	Data in

Offset	Name	Type	Description
0x10A8	GPIO_PORTC_AMUX-MODE_SET	RW	AMUXMODE
0x10C0	GPIO_PORTD_CTRL_SET	RW	Port Control
0x10C4	GPIO_PORTD_MODEL_SET	RW	Mode Low
0x10D0	GPIO_PORTD_DOUT_SET	RW	Data Out
0x10D4	GPIO_PORTD_DIN_SET	RH	Data in
0x10D8	GPIO_PORTD_AMUX-MODE_SET	RW	AMUXMODE
0x1300	GPIO_LOCK_SET	W	Lock Register
0x1310	GPIO_GPIOLOCKSTATUS_SET	RH	Lock Status
0x1320	GPIO_ABUSALLOC_SET	RW	A Bus Allocation
0x1324	GPIO_BBUSALLOC_SET	RW	B Bus Allocation
0x1328	GPIO_CDBUSALLOC_SET	RW	CD Bus Allocation
0x1400	GPIO_EXTIPSELL_SET	RW	External Interrupt Port Select Low
0x1404	GPIO_EXTIPSELH_SET	RW	External Interrupt Port Select High
0x1408	GPIO_EXTIPINSELL_SET	RW	External Interrupt Pin Select Low
0x140C	GPIO_EXTIPINSELH_SET	RW	External Interrupt Pin Select High
0x1410	GPIO_EXTIRISE_SET	RW	External Interrupt Rising Edge Trigger
0x1414	GPIO_EXTIFALL_SET	RW	External Interrupt Falling Edge Trigger
0x1420	GPIO_IF_SET	RWH INTFLAG	Interrupt Flag
0x1424	GPIO_IEN_SET	RW	Interrupt Enable
0x142C	GPIO_EM4WUEN_SET	RW	EM4 Wakeup Enable
0x1430	GPIO_EM4WUPOL_SET	RW	EM4 Wakeup Polarity
0x1440	GPIO_DBGROUTEPEN_SET	RW	Debugger Route Pin Enable
0x1444	GPIO_TRACEROUTEPEN_SET	RW	Trace Route Pin Enable
0x1450	GPIO_ACMP0_ROUTEEN_SET	RW	ACMP0 Pin Enable
0x1454	GPIO_ACMP0_ACMPOUT-ROUTE_SET	RW	ACMPOUT Port/Pin Select
0x145C	GPIO_ACMP1_ROUTEEN_SET	RW	ACMP1 Pin Enable
0x1460	GPIO_ACMP1_ACMPOUT-ROUTE_SET	RW	ACMPOUT Port/Pin Select
0x1468	GPIO_EUSART0_ROU-TEEN_SET	RW	EUSART0 Pin Enable
0x146C	GPIO_EU-SART0_CSROUTE_SET	RW	CS Port/Pin Select
0x1470	GPIO_EU-SART0_CTSROUTE_SET	RW	CTS Port/Pin Select
0x1474	GPIO_EU-SART0_RTSROUTE_SET	RW	RTS Port/Pin Select
0x1478	GPIO_EU-SART0_RXROUTE_SET	RW	RX Port/Pin Select
0x147C	GPIO_EU-SART0_SCLKROUTE_SET	RW	SCLK Port/Pin Select
0x1480	GPIO_EU-SART0_TXROUTE_SET	RW	TX Port/Pin Select

Offset	Name	Type	Description
0x1488	GPIO_EUSART1_ROUTEEN_SET	RW	EUSART1 Pin Enable
0x148C	GPIO_EUSART1_CSROUTE_SET	RW	CS Port/Pin Select
0x1490	GPIO_EUSART1_CTSROUTE_SET	RW	CTS Port/Pin Select
0x1494	GPIO_EUSART1_RTSROUTE_SET	RW	RTS Port/Pin Select
0x1498	GPIO_EUSART1_RXROUTE_SET	RW	RX Port/Pin Select
0x149C	GPIO_EUSART1_SCLKROUTE_SET	RW	SCLK Port/Pin Select
0x14A0	GPIO_EUSART1_TXROUTE_SET	RW	TX Port/Pin Select
0x14A8	GPIO_EUSART2_ROUTEEN_SET	RW	EUSART2 Pin Enable
0x14AC	GPIO_EUSART2_CSROUTE_SET	RW	CS Port/Pin Select
0x14B0	GPIO_EUSART2_CTSROUTE_SET	RW	CTS Port/Pin Select
0x14B4	GPIO_EUSART2_RTSROUTE_SET	RW	RTS Port/Pin Select
0x14B8	GPIO_EUSART2_RXROUTE_SET	RW	RX Port/Pin Select
0x14BC	GPIO_EUSART2_SCLKROUTE_SET	RW	SCLK Port/Pin Select
0x14C0	GPIO_EUSART2_TXROUTE_SET	RW	TX Port/Pin Select
0x14C8	GPIO_FRC_ROUTEEN_SET	RW	FRC Pin Enable
0x14CC	GPIO_FRC_DCLKROUTE_SET	RW	DCLK Port/Pin Select
0x14D0	GPIO_FRC_DFRAMEROUTE_SET	RW	DFRAME Port/Pin Select
0x14D4	GPIO_FRC_DOUTROUTE_SET	RW	DOUT Port/Pin Select
0x14DC	GPIO_CMU_ROUTEEN_SET	RW	HOSTCMU Pin Enable
0x14E0	GPIO_CMU_CLKIN0ROUTE_SET	RW	CLKIN0 Port/Pin Select
0x14E4	GPIO_CMU_CLKOUT0ROUTE_SET	RW	CLKOUT0 Port/Pin Select
0x14E8	GPIO_CMU_CLKOUT1ROUTE_SET	RW	CLKOUT1 Port/Pin Select
0x14EC	GPIO_CMU_CLKOUT2ROUTE_SET	RW	CLKOUT2 Port/Pin Select
0x14F8	GPIO_I2C0_ROUTEEN_SET	RW	I2C0 Pin Enable
0x14FC	GPIO_I2C0_SCLROUTE_SET	RW	SCL Port/Pin Select
0x1500	GPIO_I2C0_SDAROUTE_SET	RW	SDA Port/Pin Select
0x1508	GPIO_I2C1_ROUTEEN_SET	RW	I2C1 Pin Enable
0x150C	GPIO_I2C1_SCLROUTE_SET	RW	SCL Port/Pin Select
0x1510	GPIO_I2C1_SDAROUTE_SET	RW	SDA Port/Pin Select

Offset	Name	Type	Description
0x1518	GPIO_I2C2_ROUTEEN_SET	RW	I2C2 Pin Enable
0x151C	GPIO_I2C2_SCLROUTE_SET	RW	SCL Port/Pin Select
0x1520	GPIO_I2C2_SDAROUTE_SET	RW	SDA Port/Pin Select
0x1528	GPIO_LETIMER0_ROUTEEN_SET	RW	LETIMER0 Pin Enable
0x152C	GPIO_LETIMER0_OUT0ROUTE_SET	RW	OUT0 Port/Pin Select
0x1530	GPIO_LETIMER0_OUT1ROUTE_SET	RW	OUT1 Port/Pin Select
0x1538	GPIO_LFXO_ROUTEEN_SET	RW	LFXO Pin Enable
0x153C	GPIO_LFXO_LFXOCLKLVRAWROUTE_SET	RW	LFXOCLKLVRAW Port/Pin Select
0x1544	GPIO_MODEM_ROUTEEN_SET	RW	MODEM Pin Enable
0x1548	GPIO_MODEM_ANT0ROUTE_SET	RW	ANT0 Port/Pin Select
0x154C	GPIO_MODEM_ANT1ROUTE_SET	RW	ANT1 Port/Pin Select
0x1550	GPIO_MODEM_ANTROLLOVERROUTE_SET	RW	ANTROLLOVER Port/Pin Select
0x1554	GPIO_MODEM_ANTRR0ROUTE_SET	RW	ANTRR0 Port/Pin Select
0x1558	GPIO_MODEM_ANTRR1ROUTE_SET	RW	ANTRR1 Port/Pin Select
0x155C	GPIO_MODEM_ANTRR2ROUTE_SET	RW	ANTRR2 Port/Pin Select
0x1560	GPIO_MODEM_ANTRR3ROUTE_SET	RW	ANTRR3 Port/Pin Select
0x1564	GPIO_MODEM_ANTRR4ROUTE_SET	RW	ANTRR4 Port/Pin Select
0x1568	GPIO_MODEM_ANTRR5ROUTE_SET	RW	ANTRR5 Port/Pin Select
0x156C	GPIO_MODEM_ANTSWENROUTE_SET	RW	ANTSWEN Port/Pin Select
0x1570	GPIO_MODEM_ANTSWUSROUTE_SET	RW	ANTSWUS Port/Pin Select
0x1574	GPIO_MODEM_ANTTRIGROUTE_SET	RW	ANTTRIG Port/Pin Select
0x1578	GPIO_MODEM_ANTTRIGSTOPROUTE_SET	RW	ANTTRIGSTOP Port/Pin Select
0x157C	GPIO_MODEM_DCLKROUTE_SET	RW	DCLK Port/Pin Select
0x1580	GPIO_MODEM_DINROUTE_SET	RW	DIN Port/Pin Select
0x1584	GPIO_MODEM_DOUTROUTE_SET	RW	DOUT Port/Pin Select
0x1590	GPIO_PCNT0_S0INROUTE_SET	RW	S0IN Port/Pin Select
0x1594	GPIO_PCNT0_S1INROUTE_SET	RW	S1IN Port/Pin Select

Offset	Name	Type	Description
0x159C	GPIO_PIXELRZ0_ROUTEEN_SET	RW	PIXELRZ0 Pin Enable
0x15A0	GPIO_PIXELRZ0_RZTXOUTROUTE_SET	RW	RZTXOUT Port/Pin Select
0x15A8	GPIO_PIXELRZ1_ROUTEEN_SET	RW	PIXELRZ1 Pin Enable
0x15AC	GPIO_PIXELRZ1_RZTXOUTROUTE_SET	RW	RZTXOUT Port/Pin Select
0x15B4	GPIO_PRS0_ROUTEEN_SET	RW	PRS0 Pin Enable
0x15B8	GPIO_PRS0_ASYNCH0ROUTE_SET	RW	ASYNCH0 Port/Pin Select
0x15BC	GPIO_PRS0_ASYNCH1ROUTE_SET	RW	ASYNCH1 Port/Pin Select
0x15C0	GPIO_PRS0_ASYNCH2ROUTE_SET	RW	ASYNCH2 Port/Pin Select
0x15C4	GPIO_PRS0_ASYNCH3ROUTE_SET	RW	ASYNCH3 Port/Pin Select
0x15C8	GPIO_PRS0_ASYNCH4ROUTE_SET	RW	ASYNCH4 Port/Pin Select
0x15CC	GPIO_PRS0_ASYNCH5ROUTE_SET	RW	ASYNCH5 Port/Pin Select
0x15D0	GPIO_PRS0_ASYNCH6ROUTE_SET	RW	ASYNCH6 Port/Pin Select
0x15D4	GPIO_PRS0_ASYNCH7ROUTE_SET	RW	ASYNCH7 Port/Pin Select
0x15D8	GPIO_PRS0_ASYNCH8ROUTE_SET	RW	ASYNCH8 Port/Pin Select
0x15DC	GPIO_PRS0_ASYNCH9ROUTE_SET	RW	ASYNCH9 Port/Pin Select
0x15E0	GPIO_PRS0_ASYNCH10ROUTE_SET	RW	ASYNCH10 Port/Pin Select
0x15E4	GPIO_PRS0_ASYNCH11ROUTE_SET	RW	ASYNCH11 Port/Pin Select
0x15E8	GPIO_PRS0_SYNCH0ROUTE_SET	RW	SYNCH0 Port/Pin Select
0x15EC	GPIO_PRS0_SYNCH1ROUTE_SET	RW	SYNCH1 Port/Pin Select
0x15F0	GPIO_PRS0_SYNCH2ROUTE_SET	RW	SYNCH2 Port/Pin Select
0x15F4	GPIO_PRS0_SYNCH3ROUTE_SET	RW	SYNCH3 Port/Pin Select
0x15FC	GPIO_RAC_ROUTEEN_SET	RW	RAC Pin Enable
0x1600	GPIO_RAC_LNAENROUTE_SET	RW	LNAEN Port/Pin Select
0x1604	GPIO_RAC_PAENROUTE_SET	RW	PAEN Port/Pin Select
0x1668	GPIO_TIMER0_ROUTEEN_SET	RW	TIMER0 Pin Enable
0x166C	GPIO_TIMER0_CC0ROUTE_SET	RW	CC0 Port/Pin Select

Offset	Name	Type	Description
0x1670	GPIO_TIMER0_CC1ROUTE_SET	RW	CC1 Port/Pin Select
0x1674	GPIO_TIMER0_CC2ROUTE_SET	RW	CC2 Port/Pin Select
0x1678	GPIO_TIMER0_CDTI0ROUTE_SET	RW	CDTI0 Port/Pin Select
0x167C	GPIO_TIMER0_CDTI1ROUTE_SET	RW	CDTI1 Port/Pin Select
0x1680	GPIO_TIMER0_CDTI2ROUTE_SET	RW	CDTI2 Port/Pin Select
0x1688	GPIO_TIMER1_ROUTEEN_SET	RW	TIMER1 Pin Enable
0x168C	GPIO_TIMER1_CC0ROUTE_SET	RW	CC0 Port/Pin Select
0x1690	GPIO_TIMER1_CC1ROUTE_SET	RW	CC1 Port/Pin Select
0x1694	GPIO_TIMER1_CC2ROUTE_SET	RW	CC2 Port/Pin Select
0x1698	GPIO_TIMER1_CDTI0ROUTE_SET	RW	CDTI0 Port/Pin Select
0x169C	GPIO_TIMER1_CDTI1ROUTE_SET	RW	CDTI1 Port/Pin Select
0x16A0	GPIO_TIMER1_CDTI2ROUTE_SET	RW	CDTI2 Port/Pin Select
0x16A8	GPIO_TIMER2_ROUTEEN_SET	RW	TIMER2 Pin Enable
0x16AC	GPIO_TIMER2_CC0ROUTE_SET	RW	CC0 Port/Pin Select
0x16B0	GPIO_TIMER2_CC1ROUTE_SET	RW	CC1 Port/Pin Select
0x16B4	GPIO_TIMER2_CC2ROUTE_SET	RW	CC2 Port/Pin Select
0x16B8	GPIO_TIMER2_CC3ROUTE_SET	RW	CC3 Port/Pin Select
0x16BC	GPIO_TIMER2_CC4ROUTE_SET	RW	CC4 Port/Pin Select
0x16C0	GPIO_TIMER2_CC5ROUTE_SET	RW	CC5 Port/Pin Select
0x16C4	GPIO_TIMER2_CC6ROUTE_SET	RW	CC6 Port/Pin Select
0x16C8	GPIO_TIMER2_CDTI0ROUTE_SET	RW	CDTI0 Port/Pin Select
0x16CC	GPIO_TIMER2_CDTI1ROUTE_SET	RW	CDTI1 Port/Pin Select
0x16D0	GPIO_TIMER2_CDTI2ROUTE_SET	RW	CDTI2 Port/Pin Select
0x16D4	GPIO_TIMER2_CDTI3ROUTE_SET	RW	CDTI3 Port/Pin Select
0x16D8	GPIO_TIMER2_CDTI4ROUTE_SET	RW	CDTI4 Port/Pin Select
0x16DC	GPIO_TIMER2_CDTI5ROUTE_SET	RW	CDTI5 Port/Pin Select

Offset	Name	Type	Description
0x16E0	GPIO_TIMER2_CDTI6ROUTE_SET	RW	CDTI6 Port/Pin Select
0x16E8	GPIO_TIMER3_ROUTEEN_SET	RW	TIMER3 Pin Enable
0x16EC	GPIO_TIMER3_CC0ROUTE_SET	RW	CC0 Port/Pin Select
0x16F0	GPIO_TIMER3_CC1ROUTE_SET	RW	CC1 Port/Pin Select
0x16F4	GPIO_TIMER3_CC2ROUTE_SET	RW	CC2 Port/Pin Select
0x16F8	GPIO_TIMER3_CC3ROUTE_SET	RW	CC3 Port/Pin Select
0x16FC	GPIO_TIMER3_CC4ROUTE_SET	RW	CC4 Port/Pin Select
0x1700	GPIO_TIMER3_CC5ROUTE_SET	RW	CC5 Port/Pin Select
0x1704	GPIO_TIMER3_CC6ROUTE_SET	RW	CC6 Port/Pin Select
0x1708	GPIO_TIMER3_CDTI0ROUTE_SET	RW	CDTI0 Port/Pin Select
0x170C	GPIO_TIMER3_CDTI1ROUTE_SET	RW	CDTI1 Port/Pin Select
0x1710	GPIO_TIMER3_CDTI2ROUTE_SET	RW	CDTI2 Port/Pin Select
0x1714	GPIO_TIMER3_CDTI3ROUTE_SET	RW	CDTI3 Port/Pin Select
0x1718	GPIO_TIMER3_CDTI4ROUTE_SET	RW	CDTI4 Port/Pin Select
0x171C	GPIO_TIMER3_CDTI5ROUTE_SET	RW	CDTI5 Port/Pin Select
0x1720	GPIO_TIMER3_CDTI6ROUTE_SET	RW	CDTI6 Port/Pin Select
0x2000	GPIO_IPVERSION_CLR	R	Main
0x2030	GPIO_PORTA_CTRL_CLR	RW	Port Control
0x2034	GPIO_PORTA_MODEL_CLR	RW	Mode Low
0x203C	GPIO_PORTA_MODEH_CLR	RW	Mode High
0x2040	GPIO_PORTA_DOUT_CLR	RW	Data Out
0x2044	GPIO_PORTA_DIN_CLR	RH	Data in
0x2048	GPIO_PORTA_AMUXMODE_CLR	RW	AMUXMODE
0x2060	GPIO_PORTB_CTRL_CLR	RW	Port Control
0x2064	GPIO_PORTB_MODEL_CLR	RW	Mode Low
0x2070	GPIO_PORTB_DOUT_CLR	RW	Data Out
0x2074	GPIO_PORTB_DIN_CLR	RH	Data in
0x2078	GPIO_PORTB_AMUXMODE_CLR	RW	AMUXMODE
0x2090	GPIO_PORTC_CTRL_CLR	RW	Port Control
0x2094	GPIO_PORTC_MODEL_CLR	RW	Mode Low

Offset	Name	Type	Description
0x20A0	GPIO_PORTC_DOUT_CLR	RW	Data Out
0x20A4	GPIO_PORTC_DIN_CLR	RH	Data in
0x20A8	GPIO_PORTC_AMUX-MODE_CLR	RW	AMUXMODE
0x20C0	GPIO_PORTD_CTRL_CLR	RW	Port Control
0x20C4	GPIO_PORTD_MODEL_CLR	RW	Mode Low
0x20D0	GPIO_PORTD_DOUT_CLR	RW	Data Out
0x20D4	GPIO_PORTD_DIN_CLR	RH	Data in
0x20D8	GPIO_PORTD_AMUX-MODE_CLR	RW	AMUXMODE
0x2300	GPIO_LOCK_CLR	W	Lock Register
0x2310	GPIO_GPIOLOCKSTATUS_CLR	RH	Lock Status
0x2320	GPIO_ABUSALLOC_CLR	RW	A Bus Allocation
0x2324	GPIO_BBUSALLOC_CLR	RW	B Bus Allocation
0x2328	GPIO_CDBUSALLOC_CLR	RW	CD Bus Allocation
0x2400	GPIO_EXTIPSELL_CLR	RW	External Interrupt Port Select Low
0x2404	GPIO_EXTIPSELH_CLR	RW	External Interrupt Port Select High
0x2408	GPIO_EXTIPINSELL_CLR	RW	External Interrupt Pin Select Low
0x240C	GPIO_EXTIPINSELH_CLR	RW	External Interrupt Pin Select High
0x2410	GPIO_EXTIRISE_CLR	RW	External Interrupt Rising Edge Trigger
0x2414	GPIO_EXTIFALL_CLR	RW	External Interrupt Falling Edge Trigger
0x2420	GPIO_IF_CLR	RWH INTFLAG	Interrupt Flag
0x2424	GPIO_IEN_CLR	RW	Interrupt Enable
0x242C	GPIO_EM4WUEN_CLR	RW	EM4 Wakeup Enable
0x2430	GPIO_EM4WUPOL_CLR	RW	EM4 Wakeup Polarity
0x2440	GPIO_DBGROUTEPEN_CLR	RW	Debugger Route Pin Enable
0x2444	GPIO_TRACEROUTEPEN_CLR	RW	Trace Route Pin Enable
0x2450	GPIO_ACMP0_ROUTEEN_CLR	RW	ACMP0 Pin Enable
0x2454	GPIO_ACMP0_ACMPOUT-ROUTE_CLR	RW	ACMPOUT Port/Pin Select
0x245C	GPIO_ACMP1_ROUTEEN_CLR	RW	ACMP1 Pin Enable
0x2460	GPIO_ACMP1_ACMPOUT-ROUTE_CLR	RW	ACMPOUT Port/Pin Select
0x2468	GPIO_EUSART0_ROU-TEEN_CLR	RW	EUSART0 Pin Enable
0x246C	GPIO_EU-SART0_CSROUTE_CLR	RW	CS Port/Pin Select
0x2470	GPIO_EU-SART0_CTSROUTE_CLR	RW	CTS Port/Pin Select
0x2474	GPIO_EU-SART0_RTSROUTE_CLR	RW	RTS Port/Pin Select
0x2478	GPIO_EU-SART0_RXROUTE_CLR	RW	RX Port/Pin Select
0x247C	GPIO_EU-SART0_SCLKROUTE_CLR	RW	SCLK Port/Pin Select

Offset	Name	Type	Description
0x2480	GPIO_EUSART0_TXROUTE_CLR	RW	TX Port/Pin Select
0x2488	GPIO_EUSART1_ROUTEEN_CLR	RW	EUSART1 Pin Enable
0x248C	GPIO_EUSART1_CSROUTE_CLR	RW	CS Port/Pin Select
0x2490	GPIO_EUSART1_CTSROUTE_CLR	RW	CTS Port/Pin Select
0x2494	GPIO_EUSART1_RTSROUTE_CLR	RW	RTS Port/Pin Select
0x2498	GPIO_EUSART1_RXROUTE_CLR	RW	RX Port/Pin Select
0x249C	GPIO_EUSART1_SCLKROUTE_CLR	RW	SCLK Port/Pin Select
0x24A0	GPIO_EUSART1_TXROUTE_CLR	RW	TX Port/Pin Select
0x24A8	GPIO_EUSART2_ROUTEEN_CLR	RW	EUSART2 Pin Enable
0x24AC	GPIO_EUSART2_CSROUTE_CLR	RW	CS Port/Pin Select
0x24B0	GPIO_EUSART2_CTSROUTE_CLR	RW	CTS Port/Pin Select
0x24B4	GPIO_EUSART2_RTSROUTE_CLR	RW	RTS Port/Pin Select
0x24B8	GPIO_EUSART2_RXROUTE_CLR	RW	RX Port/Pin Select
0x24BC	GPIO_EUSART2_SCLKROUTE_CLR	RW	SCLK Port/Pin Select
0x24C0	GPIO_EUSART2_TXROUTE_CLR	RW	TX Port/Pin Select
0x24C8	GPIO_FRC_ROUTEEN_CLR	RW	FRC Pin Enable
0x24CC	GPIO_FRC_DCLKROUTE_CLR	RW	DCLK Port/Pin Select
0x24D0	GPIO_FRC_DFRAMEROUTE_CLR	RW	DFRAME Port/Pin Select
0x24D4	GPIO_FRC_DOUTROUTE_CLR	RW	DOUT Port/Pin Select
0x24DC	GPIO_CMU_ROUTEEN_CLR	RW	HOSTCMU Pin Enable
0x24E0	GPIO_CMU_CLKIN0ROUTE_CLR	RW	CLKIN0 Port/Pin Select
0x24E4	GPIO_CMU_CLKOUT0ROUTE_CLR	RW	CLKOUT0 Port/Pin Select
0x24E8	GPIO_CMU_CLKOUT1ROUTE_CLR	RW	CLKOUT1 Port/Pin Select
0x24EC	GPIO_CMU_CLKOUT2ROUTE_CLR	RW	CLKOUT2 Port/Pin Select
0x24F8	GPIO_I2C0_ROUTEEN_CLR	RW	I2C0 Pin Enable
0x24FC	GPIO_I2C0_SCLROUTE_CLR	RW	SCL Port/Pin Select
0x2500	GPIO_I2C0_SDAROUTE_CLR	RW	SDA Port/Pin Select
0x2508	GPIO_I2C1_ROUTEEN_CLR	RW	I2C1 Pin Enable

Offset	Name	Type	Description
0x250C	GPIO_I2C1_SCLROUTE_CLR	RW	SCL Port/Pin Select
0x2510	GPIO_I2C1_SDAROUTE_CLR	RW	SDA Port/Pin Select
0x2518	GPIO_I2C2_ROUTEEN_CLR	RW	I2C2 Pin Enable
0x251C	GPIO_I2C2_SCLROUTE_CLR	RW	SCL Port/Pin Select
0x2520	GPIO_I2C2_SDAROUTE_CLR	RW	SDA Port/Pin Select
0x2528	GPIO_LETIMER0_ROUTEEN_CLR	RW	LETIMER0 Pin Enable
0x252C	GPIO_LETIMER0_OUT0ROUTE_CLR	RW	OUT0 Port/Pin Select
0x2530	GPIO_LETIMER0_OUT1ROUTE_CLR	RW	OUT1 Port/Pin Select
0x2538	GPIO_LFXO_ROUTEEN_CLR	RW	LFXO Pin Enable
0x253C	GPIO_LFXO_LFXOCLKLVRAWROUTE_CLR	RW	LFXOCLKLVRAW Port/Pin Select
0x2544	GPIO_MODEM_ROUTEEN_CLR	RW	MODEM Pin Enable
0x2548	GPIO_MODEM_ANT0ROUTE_CLR	RW	ANT0 Port/Pin Select
0x254C	GPIO_MODEM_ANT1ROUTE_CLR	RW	ANT1 Port/Pin Select
0x2550	GPIO_MODEM_ANTROLLOVERROUTE_CLR	RW	ANTROLLOVER Port/Pin Select
0x2554	GPIO_MODEM_ANTRR0ROUTE_CLR	RW	ANTRR0 Port/Pin Select
0x2558	GPIO_MODEM_ANTRR1ROUTE_CLR	RW	ANTRR1 Port/Pin Select
0x255C	GPIO_MODEM_ANTRR2ROUTE_CLR	RW	ANTRR2 Port/Pin Select
0x2560	GPIO_MODEM_ANTRR3ROUTE_CLR	RW	ANTRR3 Port/Pin Select
0x2564	GPIO_MODEM_ANTRR4ROUTE_CLR	RW	ANTRR4 Port/Pin Select
0x2568	GPIO_MODEM_ANTRR5ROUTE_CLR	RW	ANTRR5 Port/Pin Select
0x256C	GPIO_MODEM_ANTSWENROUTE_CLR	RW	ANTSWEN Port/Pin Select
0x2570	GPIO_MODEM_ANTSWUSROUTE_CLR	RW	ANTSWUS Port/Pin Select
0x2574	GPIO_MODEM_ANTTRIGROUTE_CLR	RW	ANTTRIG Port/Pin Select
0x2578	GPIO_MODEM_ANTTRIGSTOPROUTE_CLR	RW	ANTTRIGSTOP Port/Pin Select
0x257C	GPIO_MODEM_DCLKROUTE_CLR	RW	DCLK Port/Pin Select
0x2580	GPIO_MODEM_DINROUTE_CLR	RW	DIN Port/Pin Select
0x2584	GPIO_MODEM_DOUTROUTE_CLR	RW	DOUT Port/Pin Select
0x2590	GPIO_PCNT0_S0INROUTE_CLR	RW	S0IN Port/Pin Select

Offset	Name	Type	Description
0x2594	GPIO_PCNT0_S1IN-ROUTE_CLR	RW	S1IN Port/Pin Select
0x259C	GPIO_PIXELRZ0_ROU-TEEN_CLR	RW	PIXELRZ0 Pin Enable
0x25A0	GPIO_PIXELRZ0_RZTXOUT-ROUTE_CLR	RW	RZTXOUT Port/Pin Select
0x25A8	GPIO_PIXELRZ1_ROU-TEEN_CLR	RW	PIXELRZ1 Pin Enable
0x25AC	GPIO_PIXELRZ1_RZTXOUT-ROUTE_CLR	RW	RZTXOUT Port/Pin Select
0x25B4	GPIO_PRS0_ROUTEEN_CLR	RW	PRS0 Pin Enable
0x25B8	GPIO_PRS0_ASYNCH0ROUTE_CLR	RW	ASYNCH0 Port/Pin Select
0x25BC	GPIO_PRS0_ASYNCH1ROUTE_CLR	RW	ASYNCH1 Port/Pin Select
0x25C0	GPIO_PRS0_ASYNCH2ROUTE_CLR	RW	ASYNCH2 Port/Pin Select
0x25C4	GPIO_PRS0_ASYNCH3ROUTE_CLR	RW	ASYNCH3 Port/Pin Select
0x25C8	GPIO_PRS0_ASYNCH4ROUTE_CLR	RW	ASYNCH4 Port/Pin Select
0x25CC	GPIO_PRS0_ASYNCH5ROUTE_CLR	RW	ASYNCH5 Port/Pin Select
0x25D0	GPIO_PRS0_ASYNCH6ROUTE_CLR	RW	ASYNCH6 Port/Pin Select
0x25D4	GPIO_PRS0_ASYNCH7ROUTE_CLR	RW	ASYNCH7 Port/Pin Select
0x25D8	GPIO_PRS0_ASYNCH8ROUTE_CLR	RW	ASYNCH8 Port/Pin Select
0x25DC	GPIO_PRS0_ASYNCH9ROUTE_CLR	RW	ASYNCH9 Port/Pin Select
0x25E0	GPIO_PRS0_ASYNCH10ROUTE_CLR	RW	ASYNCH10 Port/Pin Select
0x25E4	GPIO_PRS0_ASYNCH11ROUTE_CLR	RW	ASYNCH11 Port/Pin Select
0x25E8	GPIO_PRS0_SYNCH0ROUTE_CLR	RW	SYNCH0 Port/Pin Select
0x25EC	GPIO_PRS0_SYNCH1ROUTE_CLR	RW	SYNCH1 Port/Pin Select
0x25F0	GPIO_PRS0_SYNCH2ROUTE_CLR	RW	SYNCH2 Port/Pin Select
0x25F4	GPIO_PRS0_SYNCH3ROUTE_CLR	RW	SYNCH3 Port/Pin Select
0x25FC	GPIO_RAC_ROUTEEN_CLR	RW	RAC Pin Enable
0x2600	GPIO_RAC_LNAEN-ROUTE_CLR	RW	LNAEN Port/Pin Select
0x2604	GPIO_RAC_PAENROUTE_CLR	RW	PAEN Port/Pin Select
0x2668	GPIO_TIMER0_ROUTEEN_CLR	RW	TIMER0 Pin Enable

Offset	Name	Type	Description
0x266C	GPIO_TIMER0_CC0ROUTE_CLR	RW	CC0 Port/Pin Select
0x2670	GPIO_TIMER0_CC1ROUTE_CLR	RW	CC1 Port/Pin Select
0x2674	GPIO_TIMER0_CC2ROUTE_CLR	RW	CC2 Port/Pin Select
0x2678	GPIO_TIMER0_CDTI0ROUTE_CLR	RW	CDTI0 Port/Pin Select
0x267C	GPIO_TIMER0_CDTI1ROUTE_CLR	RW	CDTI1 Port/Pin Select
0x2680	GPIO_TIMER0_CDTI2ROUTE_CLR	RW	CDTI2 Port/Pin Select
0x2688	GPIO_TIMER1_ROUTEEN_CLR	RW	TIMER1 Pin Enable
0x268C	GPIO_TIMER1_CC0ROUTE_CLR	RW	CC0 Port/Pin Select
0x2690	GPIO_TIMER1_CC1ROUTE_CLR	RW	CC1 Port/Pin Select
0x2694	GPIO_TIMER1_CC2ROUTE_CLR	RW	CC2 Port/Pin Select
0x2698	GPIO_TIMER1_CDTI0ROUTE_CLR	RW	CDTI0 Port/Pin Select
0x269C	GPIO_TIMER1_CDTI1ROUTE_CLR	RW	CDTI1 Port/Pin Select
0x26A0	GPIO_TIMER1_CDTI2ROUTE_CLR	RW	CDTI2 Port/Pin Select
0x26A8	GPIO_TIMER2_ROUTEEN_CLR	RW	TIMER2 Pin Enable
0x26AC	GPIO_TIMER2_CC0ROUTE_CLR	RW	CC0 Port/Pin Select
0x26B0	GPIO_TIMER2_CC1ROUTE_CLR	RW	CC1 Port/Pin Select
0x26B4	GPIO_TIMER2_CC2ROUTE_CLR	RW	CC2 Port/Pin Select
0x26B8	GPIO_TIMER2_CC3ROUTE_CLR	RW	CC3 Port/Pin Select
0x26BC	GPIO_TIMER2_CC4ROUTE_CLR	RW	CC4 Port/Pin Select
0x26C0	GPIO_TIMER2_CC5ROUTE_CLR	RW	CC5 Port/Pin Select
0x26C4	GPIO_TIMER2_CC6ROUTE_CLR	RW	CC6 Port/Pin Select
0x26C8	GPIO_TIMER2_CDTI0ROUTE_CLR	RW	CDTI0 Port/Pin Select
0x26CC	GPIO_TIMER2_CDTI1ROUTE_CLR	RW	CDTI1 Port/Pin Select
0x26D0	GPIO_TIMER2_CDTI2ROUTE_CLR	RW	CDTI2 Port/Pin Select
0x26D4	GPIO_TIMER2_CDTI3ROUTE_CLR	RW	CDTI3 Port/Pin Select
0x26D8	GPIO_TIMER2_CDTI4ROUTE_CLR	RW	CDTI4 Port/Pin Select

Offset	Name	Type	Description
0x26DC	GPIO_TIMER2_CDTI5ROUTE_CLR	RW	CDTI5 Port/Pin Select
0x26E0	GPIO_TIMER2_CDTI6ROUTE_CLR	RW	CDTI6 Port/Pin Select
0x26E8	GPIO_TIMER3_ROUTEEN_CLR	RW	TIMER3 Pin Enable
0x26EC	GPIO_TIMER3_CC0ROUTE_CLR	RW	CC0 Port/Pin Select
0x26F0	GPIO_TIMER3_CC1ROUTE_CLR	RW	CC1 Port/Pin Select
0x26F4	GPIO_TIMER3_CC2ROUTE_CLR	RW	CC2 Port/Pin Select
0x26F8	GPIO_TIMER3_CC3ROUTE_CLR	RW	CC3 Port/Pin Select
0x26FC	GPIO_TIMER3_CC4ROUTE_CLR	RW	CC4 Port/Pin Select
0x2700	GPIO_TIMER3_CC5ROUTE_CLR	RW	CC5 Port/Pin Select
0x2704	GPIO_TIMER3_CC6ROUTE_CLR	RW	CC6 Port/Pin Select
0x2708	GPIO_TIMER3_CDTI0ROUTE_CLR	RW	CDTI0 Port/Pin Select
0x270C	GPIO_TIMER3_CDTI1ROUTE_CLR	RW	CDTI1 Port/Pin Select
0x2710	GPIO_TIMER3_CDTI2ROUTE_CLR	RW	CDTI2 Port/Pin Select
0x2714	GPIO_TIMER3_CDTI3ROUTE_CLR	RW	CDTI3 Port/Pin Select
0x2718	GPIO_TIMER3_CDTI4ROUTE_CLR	RW	CDTI4 Port/Pin Select
0x271C	GPIO_TIMER3_CDTI5ROUTE_CLR	RW	CDTI5 Port/Pin Select
0x2720	GPIO_TIMER3_CDTI6ROUTE_CLR	RW	CDTI6 Port/Pin Select
0x3000	GPIO_IPVERSION_TGL	R	Main
0x3030	GPIO_PORTA_CTRL_TGL	RW	Port Control
0x3034	GPIO_PORTA_MODEL_TGL	RW	Mode Low
0x303C	GPIO_PORTA_MODEH_TGL	RW	Mode High
0x3040	GPIO_PORTA_DOUT_TGL	RW	Data Out
0x3044	GPIO_PORTA_DIN_TGL	RH	Data in
0x3048	GPIO_PORTA_AMUXMODE_TGL	RW	AMUXMODE
0x3060	GPIO_PORTB_CTRL_TGL	RW	Port Control
0x3064	GPIO_PORTB_MODEL_TGL	RW	Mode Low
0x3070	GPIO_PORTB_DOUT_TGL	RW	Data Out
0x3074	GPIO_PORTB_DIN_TGL	RH	Data in
0x3078	GPIO_PORTB_AMUXMODE_TGL	RW	AMUXMODE

Offset	Name	Type	Description
0x3090	GPIO_PORTC_CTRL_TGL	RW	Port Control
0x3094	GPIO_PORTC_MODEL_TGL	RW	Mode Low
0x30A0	GPIO_PORTC_DOUT_TGL	RW	Data Out
0x30A4	GPIO_PORTC_DIN_TGL	RH	Data in
0x30A8	GPIO_PORTC_AMUX-MODE_TGL	RW	AMUXMODE
0x30C0	GPIO_PORTD_CTRL_TGL	RW	Port Control
0x30C4	GPIO_PORTD_MODEL_TGL	RW	Mode Low
0x30D0	GPIO_PORTD_DOUT_TGL	RW	Data Out
0x30D4	GPIO_PORTD_DIN_TGL	RH	Data in
0x30D8	GPIO_PORTD_AMUX-MODE_TGL	RW	AMUXMODE
0x3300	GPIO_LOCK_TGL	W	Lock Register
0x3310	GPIO_GPIOLOCKSTATUS_TGL	RH	Lock Status
0x3320	GPIO_ABUSALLOC_TGL	RW	A Bus Allocation
0x3324	GPIO_BBUSALLOC_TGL	RW	B Bus Allocation
0x3328	GPIO_CDBUSALLOC_TGL	RW	CD Bus Allocation
0x3400	GPIO_EXTIPSELL_TGL	RW	External Interrupt Port Select Low
0x3404	GPIO_EXTIPSELH_TGL	RW	External Interrupt Port Select High
0x3408	GPIO_EXTIPINSELL_TGL	RW	External Interrupt Pin Select Low
0x340C	GPIO_EXTIPINSELH_TGL	RW	External Interrupt Pin Select High
0x3410	GPIO_EXTIRISE_TGL	RW	External Interrupt Rising Edge Trigger
0x3414	GPIO_EXTIFALL_TGL	RW	External Interrupt Falling Edge Trigger
0x3420	GPIO_IF_TGL	RWH INTFLAG	Interrupt Flag
0x3424	GPIO_IEN_TGL	RW	Interrupt Enable
0x342C	GPIO_EM4WUEN_TGL	RW	EM4 Wakeup Enable
0x3430	GPIO_EM4WUPOL_TGL	RW	EM4 Wakeup Polarity
0x3440	GPIO_DBGROUTEPEN_TGL	RW	Debugger Route Pin Enable
0x3444	GPIO_TRACEROUTEPEN_TGL	RW	Trace Route Pin Enable
0x3450	GPIO_ACMP0_ROUTEEN_TGL	RW	ACMP0 Pin Enable
0x3454	GPIO_ACMP0_ACMPOUT-ROUTE_TGL	RW	ACMPOUT Port/Pin Select
0x345C	GPIO_ACMP1_ROUTEEN_TGL	RW	ACMP1 Pin Enable
0x3460	GPIO_ACMP1_ACMPOUT-ROUTE_TGL	RW	ACMPOUT Port/Pin Select
0x3468	GPIO_EUSART0_ROU-TEEN_TGL	RW	EUSART0 Pin Enable
0x346C	GPIO_EU-SART0_CSROUTE_TGL	RW	CS Port/Pin Select
0x3470	GPIO_EU-SART0_CTSROUTE_TGL	RW	CTS Port/Pin Select
0x3474	GPIO_EU-SART0_RTSROUTE_TGL	RW	RTS Port/Pin Select

Offset	Name	Type	Description
0x3478	GPIO_EUSART0_RXROUTE_TGL	RW	RX Port/Pin Select
0x347C	GPIO_EUSART0_SCLKROUTE_TGL	RW	SCLK Port/Pin Select
0x3480	GPIO_EUSART0_TXROUTE_TGL	RW	TX Port/Pin Select
0x3488	GPIO_EUSART1_ROUTEEN_TGL	RW	EUSART1 Pin Enable
0x348C	GPIO_EUSART1_CSROUTE_TGL	RW	CS Port/Pin Select
0x3490	GPIO_EUSART1_CTSROUTE_TGL	RW	CTS Port/Pin Select
0x3494	GPIO_EUSART1_RTSROUTE_TGL	RW	RTS Port/Pin Select
0x3498	GPIO_EUSART1_RXROUTE_TGL	RW	RX Port/Pin Select
0x349C	GPIO_EUSART1_SCLKROUTE_TGL	RW	SCLK Port/Pin Select
0x34A0	GPIO_EUSART1_TXROUTE_TGL	RW	TX Port/Pin Select
0x34A8	GPIO_EUSART2_ROUTEEN_TGL	RW	EUSART2 Pin Enable
0x34AC	GPIO_EUSART2_CSROUTE_TGL	RW	CS Port/Pin Select
0x34B0	GPIO_EUSART2_CTSROUTE_TGL	RW	CTS Port/Pin Select
0x34B4	GPIO_EUSART2_RTSROUTE_TGL	RW	RTS Port/Pin Select
0x34B8	GPIO_EUSART2_RXROUTE_TGL	RW	RX Port/Pin Select
0x34BC	GPIO_EUSART2_SCLKROUTE_TGL	RW	SCLK Port/Pin Select
0x34C0	GPIO_EUSART2_TXROUTE_TGL	RW	TX Port/Pin Select
0x34C8	GPIO_FRC_ROUTEEN_TGL	RW	FRC Pin Enable
0x34CC	GPIO_FRC_DCLKROUTE_TGL	RW	DCLK Port/Pin Select
0x34D0	GPIO_FRC_DFRAMEROUTE_TGL	RW	DFRAME Port/Pin Select
0x34D4	GPIO_FRC_DOUTROUTE_TGL	RW	DOUT Port/Pin Select
0x34DC	GPIO_CMU_ROUTEEN_TGL	RW	HOSTCMU Pin Enable
0x34E0	GPIO_CMU_CLKIN0ROUTE_TGL	RW	CLKIN0 Port/Pin Select
0x34E4	GPIO_CMU_CLKOUT0ROUTE_TGL	RW	CLKOUT0 Port/Pin Select
0x34E8	GPIO_CMU_CLKOUT1ROUTE_TGL	RW	CLKOUT1 Port/Pin Select
0x34EC	GPIO_CMU_CLKOUT2ROUTE_TGL	RW	CLKOUT2 Port/Pin Select
0x34F8	GPIO_I2C0_ROUTEEN_TGL	RW	I2C0 Pin Enable

Offset	Name	Type	Description
0x34FC	GPIO_I2C0_SCLROUTE_TGL	RW	SCL Port/Pin Select
0x3500	GPIO_I2C0_SDAROUTE_TGL	RW	SDA Port/Pin Select
0x3508	GPIO_I2C1_ROUTEEN_TGL	RW	I2C1 Pin Enable
0x350C	GPIO_I2C1_SCLROUTE_TGL	RW	SCL Port/Pin Select
0x3510	GPIO_I2C1_SDAROUTE_TGL	RW	SDA Port/Pin Select
0x3518	GPIO_I2C2_ROUTEEN_TGL	RW	I2C2 Pin Enable
0x351C	GPIO_I2C2_SCLROUTE_TGL	RW	SCL Port/Pin Select
0x3520	GPIO_I2C2_SDAROUTE_TGL	RW	SDA Port/Pin Select
0x3528	GPIO_LETIMER0_ROUTEEN_TGL	RW	LETIMER0 Pin Enable
0x352C	GPIO_LETIMER0_OUT0ROUTE_TGL	RW	OUT0 Port/Pin Select
0x3530	GPIO_LETIMER0_OUT1ROUTE_TGL	RW	OUT1 Port/Pin Select
0x3538	GPIO_LFXO_ROUTEEN_TGL	RW	LFXO Pin Enable
0x353C	GPIO_LFXO_LFXOCLKLVRAWROUTE_TGL	RW	LFXOCLKLVRAW Port/Pin Select
0x3544	GPIO_MODEM_ROUTEEN_TGL	RW	MODEM Pin Enable
0x3548	GPIO_MODEM_ANT0ROUTE_TGL	RW	ANT0 Port/Pin Select
0x354C	GPIO_MODEM_ANT1ROUTE_TGL	RW	ANT1 Port/Pin Select
0x3550	GPIO_MODEM_ANTROLLOVERROUTE_TGL	RW	ANTROLLOVER Port/Pin Select
0x3554	GPIO_MODEM_ANTRR0ROUTE_TGL	RW	ANTRR0 Port/Pin Select
0x3558	GPIO_MODEM_ANTRR1ROUTE_TGL	RW	ANTRR1 Port/Pin Select
0x355C	GPIO_MODEM_ANTRR2ROUTE_TGL	RW	ANTRR2 Port/Pin Select
0x3560	GPIO_MODEM_ANTRR3ROUTE_TGL	RW	ANTRR3 Port/Pin Select
0x3564	GPIO_MODEM_ANTRR4ROUTE_TGL	RW	ANTRR4 Port/Pin Select
0x3568	GPIO_MODEM_ANTRR5ROUTE_TGL	RW	ANTRR5 Port/Pin Select
0x356C	GPIO_MODEM_ANTSWENROUTE_TGL	RW	ANTSWEN Port/Pin Select
0x3570	GPIO_MODEM_ANTSWUSROUTE_TGL	RW	ANTSWUS Port/Pin Select
0x3574	GPIO_MODEM_ANTTRIGROUTE_TGL	RW	ANTTRIG Port/Pin Select
0x3578	GPIO_MODEM_ANTTRIGSTOPROUTE_TGL	RW	ANTTRIGSTOP Port/Pin Select
0x357C	GPIO_MODEM_DCLKROUTE_TGL	RW	DCLK Port/Pin Select
0x3580	GPIO_MODEM_DINROUTE_TGL	RW	DIN Port/Pin Select

Offset	Name	Type	Description
0x3584	GPIO_MODEM_DOUT-ROUTE_TGL	RW	DOUT Port/Pin Select
0x3590	GPIO_PCNT0_S0IN-ROUTE_TGL	RW	S0IN Port/Pin Select
0x3594	GPIO_PCNT0_S1IN-ROUTE_TGL	RW	S1IN Port/Pin Select
0x359C	GPIO_PIXELRZ0_ROU-TEEN_TGL	RW	PIXELRZ0 Pin Enable
0x35A0	GPIO_PIXELRZ0_RZTXOUT-ROUTE_TGL	RW	RZTXOUT Port/Pin Select
0x35A8	GPIO_PIXELRZ1_ROU-TEEN_TGL	RW	PIXELRZ1 Pin Enable
0x35AC	GPIO_PIXELRZ1_RZTXOUT-ROUTE_TGL	RW	RZTXOUT Port/Pin Select
0x35B4	GPIO_PRS0_ROUTEEN_TGL	RW	PRS0 Pin Enable
0x35B8	GPIO_PRS0_ASYNCH0ROUTE_TGL	RW	ASYNCH0 Port/Pin Select
0x35BC	GPIO_PRS0_ASYNCH1ROUTE_TGL	RW	ASYNCH1 Port/Pin Select
0x35C0	GPIO_PRS0_ASYNCH2ROUTE_TGL	RW	ASYNCH2 Port/Pin Select
0x35C4	GPIO_PRS0_ASYNCH3ROUTE_TGL	RW	ASYNCH3 Port/Pin Select
0x35C8	GPIO_PRS0_ASYNCH4ROUTE_TGL	RW	ASYNCH4 Port/Pin Select
0x35CC	GPIO_PRS0_ASYNCH5ROUTE_TGL	RW	ASYNCH5 Port/Pin Select
0x35D0	GPIO_PRS0_ASYNCH6ROUTE_TGL	RW	ASYNCH6 Port/Pin Select
0x35D4	GPIO_PRS0_ASYNCH7ROUTE_TGL	RW	ASYNCH7 Port/Pin Select
0x35D8	GPIO_PRS0_ASYNCH8ROUTE_TGL	RW	ASYNCH8 Port/Pin Select
0x35DC	GPIO_PRS0_ASYNCH9ROUTE_TGL	RW	ASYNCH9 Port/Pin Select
0x35E0	GPIO_PRS0_ASYNCH10ROUTE_TGL	RW	ASYNCH10 Port/Pin Select
0x35E4	GPIO_PRS0_ASYNCH11ROUTE_TGL	RW	ASYNCH11 Port/Pin Select
0x35E8	GPIO_PRS0_SYNCH0ROUTE_TGL	RW	SYNCH0 Port/Pin Select
0x35EC	GPIO_PRS0_SYNCH1ROUTE_TGL	RW	SYNCH1 Port/Pin Select
0x35F0	GPIO_PRS0_SYNCH2ROUTE_TGL	RW	SYNCH2 Port/Pin Select
0x35F4	GPIO_PRS0_SYNCH3ROUTE_TGL	RW	SYNCH3 Port/Pin Select
0x35FC	GPIO_RAC_ROUTEEN_TGL	RW	RAC Pin Enable
0x3600	GPIO_RAC_LNAEN-ROUTE_TGL	RW	LNAEN Port/Pin Select

Offset	Name	Type	Description
0x3604	GPIO_RAC_PAENROUTE_TGL	RW	PAEN Port/Pin Select
0x3668	GPIO_TIMER0_ROUTEEN_TGL	RW	TIMER0 Pin Enable
0x366C	GPIO_TIMER0_CC0ROUTE_TGL	RW	CC0 Port/Pin Select
0x3670	GPIO_TIMER0_CC1ROUTE_TGL	RW	CC1 Port/Pin Select
0x3674	GPIO_TIMER0_CC2ROUTE_TGL	RW	CC2 Port/Pin Select
0x3678	GPIO_TIMER0_CDTI0ROUTE_TGL	RW	CDTI0 Port/Pin Select
0x367C	GPIO_TIMER0_CDTI1ROUTE_TGL	RW	CDTI1 Port/Pin Select
0x3680	GPIO_TIMER0_CDTI2ROUTE_TGL	RW	CDTI2 Port/Pin Select
0x3688	GPIO_TIMER1_ROUTEEN_TGL	RW	TIMER1 Pin Enable
0x368C	GPIO_TIMER1_CC0ROUTE_TGL	RW	CC0 Port/Pin Select
0x3690	GPIO_TIMER1_CC1ROUTE_TGL	RW	CC1 Port/Pin Select
0x3694	GPIO_TIMER1_CC2ROUTE_TGL	RW	CC2 Port/Pin Select
0x3698	GPIO_TIMER1_CDTI0ROUTE_TGL	RW	CDTI0 Port/Pin Select
0x369C	GPIO_TIMER1_CDTI1ROUTE_TGL	RW	CDTI1 Port/Pin Select
0x36A0	GPIO_TIMER1_CDTI2ROUTE_TGL	RW	CDTI2 Port/Pin Select
0x36A8	GPIO_TIMER2_ROUTEEN_TGL	RW	TIMER2 Pin Enable
0x36AC	GPIO_TIMER2_CC0ROUTE_TGL	RW	CC0 Port/Pin Select
0x36B0	GPIO_TIMER2_CC1ROUTE_TGL	RW	CC1 Port/Pin Select
0x36B4	GPIO_TIMER2_CC2ROUTE_TGL	RW	CC2 Port/Pin Select
0x36B8	GPIO_TIMER2_CC3ROUTE_TGL	RW	CC3 Port/Pin Select
0x36BC	GPIO_TIMER2_CC4ROUTE_TGL	RW	CC4 Port/Pin Select
0x36C0	GPIO_TIMER2_CC5ROUTE_TGL	RW	CC5 Port/Pin Select
0x36C4	GPIO_TIMER2_CC6ROUTE_TGL	RW	CC6 Port/Pin Select
0x36C8	GPIO_TIMER2_CDTI0ROUTE_TGL	RW	CDTI0 Port/Pin Select
0x36CC	GPIO_TIMER2_CDTI1ROUTE_TGL	RW	CDTI1 Port/Pin Select
0x36D0	GPIO_TIMER2_CDTI2ROUTE_TGL	RW	CDTI2 Port/Pin Select

Offset	Name	Type	Description
0x36D4	GPIO_TIMER2_CDTI3ROUTE_TGL	RW	CDTI3 Port/Pin Select
0x36D8	GPIO_TIMER2_CDTI4ROUTE_TGL	RW	CDTI4 Port/Pin Select
0x36DC	GPIO_TIMER2_CDTI5ROUTE_TGL	RW	CDTI5 Port/Pin Select
0x36E0	GPIO_TIMER2_CDTI6ROUTE_TGL	RW	CDTI6 Port/Pin Select
0x36E8	GPIO_TIMER3_ROUTEEN_TGL	RW	TIMER3 Pin Enable
0x36EC	GPIO_TIMER3_CC0ROUTE_TGL	RW	CC0 Port/Pin Select
0x36F0	GPIO_TIMER3_CC1ROUTE_TGL	RW	CC1 Port/Pin Select
0x36F4	GPIO_TIMER3_CC2ROUTE_TGL	RW	CC2 Port/Pin Select
0x36F8	GPIO_TIMER3_CC3ROUTE_TGL	RW	CC3 Port/Pin Select
0x36FC	GPIO_TIMER3_CC4ROUTE_TGL	RW	CC4 Port/Pin Select
0x3700	GPIO_TIMER3_CC5ROUTE_TGL	RW	CC5 Port/Pin Select
0x3704	GPIO_TIMER3_CC6ROUTE_TGL	RW	CC6 Port/Pin Select
0x3708	GPIO_TIMER3_CDTI0ROUTE_TGL	RW	CDTI0 Port/Pin Select
0x370C	GPIO_TIMER3_CDTI1ROUTE_TGL	RW	CDTI1 Port/Pin Select
0x3710	GPIO_TIMER3_CDTI2ROUTE_TGL	RW	CDTI2 Port/Pin Select
0x3714	GPIO_TIMER3_CDTI3ROUTE_TGL	RW	CDTI3 Port/Pin Select
0x3718	GPIO_TIMER3_CDTI4ROUTE_TGL	RW	CDTI4 Port/Pin Select
0x371C	GPIO_TIMER3_CDTI5ROUTE_TGL	RW	CDTI5 Port/Pin Select
0x3720	GPIO_TIMER3_CDTI6ROUTE_TGL	RW	CDTI6 Port/Pin Select

25.6 GPIO Register Description

25.6.1 GPIO_IPVERSION - Main

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x8																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x8	R	**ip version id**
	IPVERSION ID			

25.6.2 GPIO_PORTA_CTRL - Port Control

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0						0x4										0x0						0x4						
Access				RW						RW										RW						RW						
Name				DINDISALT						SLEWRATEALT										DINDIS						SLEWRATE						

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	DINDISALT	0x0	RW	**Data In Disable Alt**
	Data input disable for port pins using alternate modes.			
27:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	SLEWRATEALT	0x4	RW	**Slew Rate Alt**
	Slewrate limit for port pins using alternate modes. Higher values provide faster slewrates.			
19:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	DINDIS	0x0	RW	**Data In Disable**
	Data input disable for port pins not using alternate modes.			
11:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	SLEWRATE	0x4	RW	**Slew Rate**
	Slewrate limit for port pins using not alternate modes. Higher values provide faster slewrates.			
3:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

25.6.3 GPIO_PORTA_MODEL - Mode Low

Offset	Bit Position																															
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0			
Access	RW				RW				RW				RW				RW				RW				RW				RW			
Name	MODE7				MODE6				MODE5				MODE4				MODE3				MODE2				MODE1				MODE0			

Bit	Name	Reset	Access	Description
31:28	MODE7	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

Bit	Name	Reset	Access	Description
27:24	MODE6	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.

Bit	Name	Reset	Access	Description
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
23:20	MODE5 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
19:16	MODE4 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.

Bit	Name	Reset	Access	Description
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
15:12	MODE3 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
11:8	MODE2 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.

Bit	Name	Reset	Access	Description
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
7:4	MODE1 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
3:0	MODE0 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.

Bit	Name	Reset	Access	Description
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.

25.6.4 GPIO_PORTA_MODEH - Mode High

Offset	Bit Position																															
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													MODE0			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	MODE0 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

25.6.5 GPIO_PORTA_DOUT - Data Out

Offset	Bit Position																														
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0								
Access																								RW								
Name																								DOUT								

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	DOUT Data output	0x0	RW	**Data output**

25.6.6 GPIO_PORTA_DIN - Data in

Offset	Bit Position																														
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0								
Access																								R								
Name																								DIN								

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	DIN Data input	0x0	R	**Data input**

25.6.7 GPIO_PORTA_AMUXMODE - AMUXMODE

Offset	Bit Position																														
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																								0x0								
Access																								RW								
Name																								LNEN								

Bit	Name	Reset	Access	Description
31:9	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
8:0	LNEN Enable low-noise mode	0x0	RW	**Low Noise Enable**

25.6.8 GPIO_PORTB_CTRL - Port Control

Offset	Bit Position																													
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0						0x4										0x0						0x4						
Access				RW						RW										RW						RW						
Name				DINDISALT						SLEWRATEALT										DINDIS						SLEWRATE						

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	DINDISALT	0x0	RW	**Data In Disable Alt**
	Data input disable for port pins using alternate modes.			
27:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	SLEWRATEALT	0x4	RW	**Slew Rate Alt**
	Slewrate limit for port pins using alternate modes. Higher values provide faster slewrates.			
19:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	DINDIS	0x0	RW	**Data In Disable**
	Data input disable for port pins not using alternate modes.			
11:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	SLEWRATE	0x4	RW	**Slew Rate**
	Slewrate limit for port pins using not alternate modes. Higher values provide faster slewrates.			
3:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

25.6.9 GPIO_PORTB_MODEL - Mode Low

Offset	Bit Position																													
0x064	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0				0x0				0x0				0x0			
Access																	RW				RW				RW				RW			
Name																	MODE3				MODE2				MODE1				MODE0			

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:12	MODE3 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

11:8	MODE2 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.

Bit	Name	Reset	Access	Description
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
7:4	MODE1 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
3:0	MODE0 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.

Bit	Name	Reset	Access	Description
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.

25.6.10 GPIO_PORTB_DOUT - Data Out

Offset	Bit Position																															
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													DOUT			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	DOUT Data output	0x0	RW	**Data output**

25.6.11 GPIO_PORTB_DIN - Data in

Offset	Bit Position																															
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													R			
Name																													DIN			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	DIN Data input	0x0	R	**Data input**

25.6.12 GPIO_PORTB_AMUXMODE - AMUXMODE

Offset	Bit Position																															
0x078	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0			
Access																													RW			
Name																													LNEN			

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	LNEN	0x0	RW	**Low Noise Enable**
	Enable low-noise mode			

25.6.13 GPIO_PORTC_CTRL - Port Control

Offset	Bit Position																															
0x090	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0						0x4										0x0						0x4						
Access				RW						RW										RW						RW						
Name				DINDISALT						SLEWRATEALT										DINDIS						SLEWRATE						

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	DINDISALT	0x0	RW	**Data In Disable Alt**
	Data input disable for port pins using alternate modes.			
27:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	SLEWRATEALT	0x4	RW	**Slew Rate Alt**
	Slewrate limit for port pins using alternate modes. Higher values provide faster slewrates.			
19:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	DINDIS	0x0	RW	**Data In Disable**
	Data input disable for port pins not using alternate modes.			
11:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	SLEWRATE	0x4	RW	**Slew Rate**
	Slewrate limit for port pins using not alternate modes. Higher values provide faster slewrates.			
3:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

25.6.14 GPIO_PORTC_MODEL - Mode Low

Offset	Bit Position																													
0x094	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0			
Access	RW				RW				RW				RW				RW				RW				RW				RW			
Name	MODE7				MODE6				MODE5				MODE4				MODE3				MODE2				MODE1				MODE0			

Bit	Name	Reset	Access	Description
31:28	MODE7	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

Bit	Name	Reset	Access	Description
27:24	MODE6	0x0	RW	**MODE n**
	MODE n			

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.

Bit	Name	Reset	Access	Description
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
23:20	MODE5 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
19:16	MODE4 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.

Bit	Name	Reset	Access	Description
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
15:12	MODE3 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
11:8	MODE2 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.

Bit	Name	Reset	Access	Description
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
7:4	MODE1 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
3:0	MODE0 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.

Bit	Name	Reset	Access	Description
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.

25.6.15 GPIO_PORTC_DOUT - Data Out

Offset	Bit Position																													
0x0A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									DOUT							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	DOUT	0x0	RW	**Data output**
	Data output			

25.6.16 GPIO_PORTC_DIN - Data in

Offset	Bit Position																													
0x0A4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									DIN							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	DIN	0x0	R	**Data input**
	Data input			

25.6.17 GPIO_PORTC_AMUXMODE - AMUXMODE

Offset	Bit Position																													
0x0A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									LNEN							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	LNEN	0x0	RW	**Low Noise Enable**
	Enable low-noise mode			

25.6.18 GPIO_PORTD_CTRL - Port Control

Offset	Bit Position																													
0x0C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0						0x4										0x0						0x4						
Access				RW						RW										RW						RW						
Name				DINDISALT						SLEWRATEALT										DINDIS						SLEWRATE						

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28	DINDISALT	0x0	RW	**Data In Disable Alt**
	Data input disable for port pins using alternate modes.			
27:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22:20	SLEWRATEALT	0x4	RW	**Slew Rate Alt**
	Slewrate limit for port pins using alternate modes. Higher values provide faster slewrates.			
19:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	DINDIS	0x0	RW	**Data In Disable**
	Data input disable for port pins not using alternate modes.			
11:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:4	SLEWRATE	0x4	RW	**Slew Rate**
	Slewrate limit for port pins using not alternate modes. Higher values provide faster slewrates.			
3:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

25.6.19 GPIO_PORTD_MODEL - Mode Low

Offset	Bit Position																													
0x0C4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0				0x0				0x0				0x0				0x0				0x0				0x0			
Access					RW				RW				RW				RW				RW				RW				RW			
Name					MODE6				MODE5				MODE4				MODE3				MODE2				MODE1				MODE0			

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	MODE6 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

23:20	MODE5 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.

Bit	Name	Reset	Access	Description
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
19:16	MODE4 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
15:12	MODE3 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.

Bit	Name	Reset	Access	Description
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
11:8	MODE2 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
7:4	MODE1 MODE n	0x0	RW	**MODE n**
	Value	Mode		Description
	0	DISABLED		Input disabled. Pullup if DOUT is set.
	1	INPUT		Input enabled. Filter if DOUT is set.
	2	INPUTPULL		Input enabled. DOUT determines pull direction.
	3	INPUTPULLFILTER		Input enabled with filter. DOUT determines pull direction.

Bit	Name	Reset	Access	Description
	4	PUSHPULL		Push-pull output.
	5	PUSHPULLALT		Push-pull using alternate control.
	6	WIREDOR		Wired-or output.
	7	WIREDORPULLDOWN		Wired-or output with pull-down.
	8	WIREDAND		Open-drain output.
	9	WIREDANDFILTER		Open-drain output with filter.
	10	WIREDANDPULLUP		Open-drain output with pullup.
	11	WIREDANDPULLUPFIL-TER		Open-drain output with filter and pullup.
	12	WIREDANDALT		Open-drain output using alternate control.
	13	WIREDANDALTFILTER		Open-drain output using alternate control with filter.
	14	WIREDANDALTPULLUP		Open-drain output using alternate control with pullup.
	15	WIREDANDALTPULLUP-FILTER		Open-drain output using alternate control with filter and pullup.
3:0	MODE0 MODE n	0x0	RW	**MODE n**

Value	Mode	Description
0	DISABLED	Input disabled. Pullup if DOUT is set.
1	INPUT	Input enabled. Filter if DOUT is set.
2	INPUTPULL	Input enabled. DOUT determines pull direction.
3	INPUTPULLFILTER	Input enabled with filter. DOUT determines pull direction.
4	PUSHPULL	Push-pull output.
5	PUSHPULLALT	Push-pull using alternate control.
6	WIREDOR	Wired-or output.
7	WIREDORPULLDOWN	Wired-or output with pull-down.
8	WIREDAND	Open-drain output.
9	WIREDANDFILTER	Open-drain output with filter.
10	WIREDANDPULLUP	Open-drain output with pullup.
11	WIREDANDPULLUPFIL-TER	Open-drain output with filter and pullup.
12	WIREDANDALT	Open-drain output using alternate control.
13	WIREDANDALTFILTER	Open-drain output using alternate control with filter.
14	WIREDANDALTPULLUP	Open-drain output using alternate control with pullup.
15	WIREDANDALTPULLUP-FILTER	Open-drain output using alternate control with filter and pullup.

25.6.20 GPIO_PORTD_DOUT - Data Out

Offset	Bit Position																													
0x0D0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																										0x0						
Access																										RW						
Name																										DOUT						

Bit	Name	Reset	Access	Description
31:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:0	DOUT Data output	0x0	RW	**Data output**

25.6.21 GPIO_PORTD_DIN - Data in

Offset	Bit Position																													
0x0D4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																										0x0						
Access																										R						
Name																										DIN						

Bit	Name	Reset	Access	Description
31:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:0	DIN Data input	0x0	R	**Data input**

25.6.22 GPIO_PORTD_AMUXMODE - AMUXMODE

Offset	Bit Position																													
0x0D8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																										0x0						
Access																										RW						
Name																										LNEN						

Bit	Name	Reset	Access	Description
31:7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:0	LNEN Enable low-noise mode	0x0	RW	**Low Noise Enable**

25.6.23 GPIO_LOCK - Lock Register

Offset	Bit Position																													
0x300	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0xA534															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0xA534	W	**Configuration Lock Key**

Write any other value than the unlock code to lock configuration registers. Write the unlock code to unlock (See text for detailed list of configuration registers.)

Value	Mode	Description
42292	UNLOCK	Unlock code

25.6.24 GPIO_GPIOLOCKSTATUS - Lock Status

Offset	Bit Position																													
0x310	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																LOCK

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	LOCK	0x0	R	**GPIO LOCK status**

Indicates current lock status of GPIO registers

Value	Mode	Description
0	UNLOCKED	Registers are unlocked
1	LOCKED	Registers are locked

25.6.25 GPIO_ABUSALLOC - A Bus Allocation

Offset	Bit Position																													
0x320	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0								0x0								0x0								0x0			
Access					RW								RW								RW								RW			
Name					AODD1								AODD0								AEVEN1								AEVEN0			

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	AODD1	0x0	RW	**A Bus Odd 1**
	peripheral allocation to A Bus Odd 1			

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1

23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	AODD0	0x0	RW	**A Bus Odd 0**
	peripheral allocation to A Bus Odd 0			

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1

15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	AEVEN1	0x0	RW	**A Bus Even 1**
	peripheral allocation to A Bus Even 1			

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1

7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	AEVEN0	0x0	RW	**A Bus Even 0**
	peripheral allocation to A Bus Even 0			

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	TRISTATE		The bus is not allocated
	1	ADC0		The bus is allocated to ADC0
	2	ACMP0		The bus is allocated to ACMP0
	3	ACMP1		The bus is allocated to ACMP1

25.6.26 GPIO_BBUSALLOC - B Bus Allocation

Offset	Bit Position																													
0x324	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0								0x0								0x0								0x0			
Access					RW								RW								RW								RW			
Name					BODD1								BODD0								BEVEN1								BEVEN0			

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	BODD1	0x0	RW	**B Bus Odd 1**
	peripheral allocation to B Bus Odd 1			

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	VDAC0CH1	The bus is allocated to VDAC0 CH1
5	VDAC1CH1	The bus is allocated to VDAC1 CH1

Bit	Name	Reset	Access	Description
23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	BODD0	0x0	RW	**B Bus Odd 0**
	peripheral allocation to B Bus Odd 0			

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	VDAC0CH0	The bus is allocated to VDAC0 CH0
5	VDAC1CH0	The bus is allocated to VDAC1 CH0

Bit	Name	Reset	Access	Description
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	BEVEN1	0x0	RW	**B Bus Even 1**
	peripheral allocation to B Bus Even 1			

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1

Bit	Name	Reset	Access	Description
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	BEVEN0	0x0	RW	**B Bus Even 0**

peripheral allocation to B Bus Even 0

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
14	DIAGA	The bus is allocated to DIAGA

25.6.27 GPIO_CDBUSALLOC - CD Bus Allocation

Offset	Bit Position																													
0x328	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0								0x0								0x0								0x0			
Access					RW								RW								RW								RW			
Name					CDODD1								CDODD0								CDEVEN1								CDEVEN0			

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:24	CDODD1	0x0	RW	**CD Bus Odd 1**
	peripheral allocation to CD Bus Odd 1			

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	LEDDRV	The bus is allocated to LEDDRV

Bit	Name	Reset	Access	Description
23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	CDODD0	0x0	RW	**CD Bus Odd 0**
	peripheral allocation to CD Bus Odd 0			

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	LEDDRV	The bus is allocated to LEDDRV

Bit	Name	Reset	Access	Description
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	CDEVEN1	0x0	RW	**CD Bus Even 1**
	peripheral allocation to CD Bus Even 1			

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	LEDDRV	The bus is allocated to LEDDRV

Bit	Name	Reset	Access	Description
7:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	CDEVEN0	0x0	RW	**CD Bus Even 0** peripheral allocation to CD Bus Even 0

Value	Mode	Description
0	TRISTATE	The bus is not allocated
1	ADC0	The bus is allocated to ADC0
2	ACMP0	The bus is allocated to ACMP0
3	ACMP1	The bus is allocated to ACMP1
4	LEDDRV	The bus is allocated to LEDDRV

25.6.28 GPIO_EXTIPSELL - External Interrupt Port Select Low

Offset	Bit Position																													
0x400	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0	
Access			RW				RW				RW				RW				RW				RW				RW				RW	
Name			EXTIPSEL7				EXTIPSEL6				EXTIPSEL5				EXTIPSEL4				EXTIPSEL3				EXTIPSEL2				EXTIPSEL1				EXTIPSEL0	

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29:28	EXTIPSEL7	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 7 (EXTI7)			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
27:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:24	EXTIPSEL6	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 6 (EXTI6)			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
23:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:20	EXTIPSEL5	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 5 (EXTI5)			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	EXTIPSEL4	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 4 (EXTI4)			

Bit	Name	Reset	Access	Description
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	EXTIPSEL3	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 3 (EXTI3)			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	EXTIPSEL2	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 2 (EXTI2)			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	EXTIPSEL1	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 1 (EXTI1)			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	EXTIPSEL0	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 0 (EXTI0)			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected

Bit	Name	Reset	Access	Description
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected

25.6.29 GPIO_EXTIPSELH - External Interrupt Port Select High

Offset	Bit Position																													
0x404	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																			0x0				0x0				0x0				0x0	
Access																			RW				RW				RW				RW	
Name																			EXTIPSEL3				EXTIPSEL2				EXTIPSEL1				EXTIPSEL0	

Bit	Name	Reset	Access	Description
31:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	EXTIPSEL3	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 3+8			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	EXTIPSEL2	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 2+8			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	EXTIPSEL1	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 1+8			
	Value	Mode		Description
	0	PORTA		Port A group selected
	1	PORTB		Port B group selected
	2	PORTC		Port C group selected
	3	PORTD		Port D group selected
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	EXTIPSEL0	0x0	RW	**External Interrupt Port Select**
	Port select for external interrupt 0+8			

Bit	Name	Reset	Access	Description

Value	Mode	Description
0	PORTA	Port A group selected
1	PORTB	Port B group selected
2	PORTC	Port C group selected
3	PORTD	Port D group selected

25.6.30 GPIO_EXTIPINSELL - External Interrupt Pin Select Low

Offset	Bit Position																													
0x408	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x0				0x0				0x0				0x0				0x0				0x0				0x0				0x0	
Access			RW				RW				RW				RW				RW				RW				RW				RW	
Name			EXTIPINSEL7				EXTIPINSEL6				EXTIPINSEL5				EXTIPINSEL4				EXTIPINSEL3				EXTIPINSEL2				EXTIPINSEL1				EXTIPINSEL0	

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29:28	EXTIPINSEL7	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 7 (EXTI7)			
	Value	Mode		Description
	0	PIN0		OFFSET=0
	1	PIN1		OFFSET=1
	2	PIN2		OFFSET=2
	3	PIN3		OFFSET=3
27:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:24	EXTIPINSEL6	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 6 (EXTI6)			
	Value	Mode		Description
	0	PIN0		OFFSET=0
	1	PIN1		OFFSET=1
	2	PIN2		OFFSET=2
	3	PIN3		OFFSET=3
23:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:20	EXTIPINSEL5	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 5 (EXTI5)			
	Value	Mode		Description
	0	PIN0		OFFSET=0
	1	PIN1		OFFSET=1
	2	PIN2		OFFSET=2
	3	PIN3		OFFSET=3
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	EXTIPINSEL4	0x0	RW	**External Interrupt Pin select**

Bit	Name	Reset	Access	Description
	OFFSET select for External Interrupt 4 (EXTI4)			

Value	Mode	Description
0	PIN0	OFFSET=0
1	PIN1	OFFSET=1
2	PIN2	OFFSET=2
3	PIN3	OFFSET=3

15:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	EXTIPINSEL3	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 3 (EXTI3)			

Value	Mode	Description
0	PIN0	OFFSET=0
1	PIN1	OFFSET=1
2	PIN2	OFFSET=2
3	PIN3	OFFSET=3

11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	EXTIPINSEL2	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 2 (EXTI2)			

Value	Mode	Description
0	PIN0	OFFSET=0
1	PIN1	OFFSET=1
2	PIN2	OFFSET=2
3	PIN3	OFFSET=3

7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	EXTIPINSEL1	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 1 (EXTI1)			

Value	Mode	Description
0	PIN0	OFFSET=0
1	PIN1	OFFSET=1
2	PIN2	OFFSET=2
3	PIN3	OFFSET=3

3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	EXTIPINSEL0	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt 0 (EXTI0)			

Value	Mode	Description
0	PIN0	OFFSET=0

Bit	Name	Reset	Access	Description
	1	PIN1		OFFSET=1
	2	PIN2		OFFSET=2
	3	PIN3		OFFSET=3

25.6.31 GPIO_EXTIPINSELH - External Interrupt Pin Select High

Offset	Bit Position																													
0x40C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																			0x0				0x0				0x0				0x0	
Access																			RW				RW				RW				RW	
Name																			EXTIPINSEL3				EXTIPINSEL2				EXTIPINSEL1				EXTIPINSEL0	

Bit	Name	Reset	Access	Description
31:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13:12	EXTIPINSEL3	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt {b+8} (EXTI{b+8})			

Value	Mode	Description
0	PIN8	OFFSET=8
1	PIN9	OFFSET=9
2	PIN10	OFFSET=10
3	PIN11	OFFSET=11

11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	EXTIPINSEL2	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt {b+8} (EXTI{b+8})			

Value	Mode	Description
0	PIN8	OFFSET=8
1	PIN9	OFFSET=9
2	PIN10	OFFSET=10
3	PIN11	OFFSET=11

7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5:4	EXTIPINSEL1	0x0	RW	**External Interrupt Pin select**
	OFFSET select for External Interrupt {b+8} (EXTI{b+8})			

Value	Mode	Description
0	PIN8	OFFSET=8
1	PIN9	OFFSET=9
2	PIN10	OFFSET=10
3	PIN11	OFFSET=11

3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	EXTIPINSEL0	0x0	RW	**External Interrupt Pin select**

Bit	Name	Reset	Access	Description
	OFFSET select for External Interrupt {b+8} (EXTI{b+8})			

Value	Mode	Description
0	PIN8	OFFSET=8
1	PIN9	OFFSET=9
2	PIN10	OFFSET=10
3	PIN11	OFFSET=11

25.6.32 GPIO_EXTIRISE - External Interrupt Rising Edge Trigger

Offset	Bit Position																															
0x410	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0											
Access																					RW											
Name																					EXTIRISE											

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:0	EXTIRISE	0x0	RW	**EXT Int Rise**
	External Interrupt n Rising Edge Trigger Enable			

25.6.33 GPIO_EXTIFALL - External Interrupt Falling Edge Trigger

Offset	Bit Position																															
0x414	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0											
Access																					RW											
Name																					EXTIFALL											

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:0	EXTIFALL	0x0	RW	**EXT Int FALL**
	External Interrupt n Falling Edge Trigger Enable			

25.6.34 GPIO_IF - Interrupt Flag

Offset	Bit Position																													
0x420	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name					EM4WUIF11	EM4WUIF10	EM4WUIF9	EM4WUIF8	EM4WUIF7	EM4WUIF6	EM4WUIF5	EM4WUIF4	EM4WUIF3	EM4WUIF2	EM4WUIF1	EM4WUIF0					EXTIF11	EXTIF10	EXTIF9	EXTIF8	EXTIF7	EXTIF6	EXTIF5	EXTIF4	EXTIF3	EXTIF2	EXTIF1	EXTIF0

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	EM4WUIF11	0x0	RW	**EM4 wake up**
	EM4 wakeup interrupt flag			
26	EM4WUIF10	0x0	RW	**EM4 wake up**
	EM4 wakeup interrupt flag			
25	EM4WUIF9	0x0	RW	**EM4 wake up**
	EM4 wakeup interrupt flag			
24	EM4WUIF8	0x0	RW	**EM4 wake up**
	EM4 wakeup interrupt flag			
23	EM4WUIF7	0x0	RW	**EM4 wake up**
	EM4 wakeup interrupt flag			
22	EM4WUIF6	0x0	RW	**EM4 wake up**
	EM4 wakeup interrupt flag			
21	EM4WUIF5	0x0	RW	**EM4 wake up**
	EM4 wakeup interrupt flag			
20	EM4WUIF4	0x0	RW	**EM4 wake up**
	EM4 wakeup interrupt flag			
19	EM4WUIF3	0x0	RW	**EM4 wake up**
	EM4 wakeup interrupt flag			
18	EM4WUIF2	0x0	RW	**EM4 wake up**
	EM4 wakeup interrupt flag			
17	EM4WUIF1	0x0	RW	**EM4 wake up**
	EM4 wakeup interrupt flag			
16	EM4WUIF0	0x0	RW	**EM4 wake up**
	EM4 wakeup interrupt flag			
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	EXTIF11	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			
10	EXTIF10	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			

Bit	Name	Reset	Access	Description
9	EXTIF9	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			
8	EXTIF8	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			
7	EXTIF7	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			
6	EXTIF6	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			
5	EXTIF5	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			
4	EXTIF4	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			
3	EXTIF3	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			
2	EXTIF2	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			
1	EXTIF1	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			
0	EXTIF0	0x0	RW	**External Pin Flag**
	External Pin interrupt flag			

25.6.35 GPIO_IEN - Interrupt Enable

Offset	Bit Position																													
0x424	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0					0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name					EM4WUIEN11	EM4WUIEN10	EM4WUIEN9	EM4WUIEN8	EM4WUIEN7	EM4WUIEN6	EM4WUIEN5	EM4WUIEN4	EM4WUIEN3	EM4WUIEN2	EM4WUIEN1	EM4WUIEN0					EXTIEN11	EXTIEN10	EXTIEN9	EXTIEN8	EXTIEN7	EXTIEN6	EXTIEN5	EXTIEN4	EXTIEN3	EXTIEN2	EXTIEN1	EXTIEN0

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27	EM4WUIEN11	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
26	EM4WUIEN10	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
25	EM4WUIEN9	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
24	EM4WUIEN8	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
23	EM4WUIEN7	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
22	EM4WUIEN6	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
21	EM4WUIEN5	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
20	EM4WUIEN4	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
19	EM4WUIEN3	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
18	EM4WUIEN2	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
17	EM4WUIEN1	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
16	EM4WUIEN0	0x0	RW	**EM4 Wake Up Interrupt En**
	EM4 wakeup interrupt enable			
15:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	EXTIEN11	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
10	EXTIEN10	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			

Bit	Name	Reset	Access	Description
9	EXTIEN9	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
8	EXTIEN8	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
7	EXTIEN7	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
6	EXTIEN6	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
5	EXTIEN5	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
4	EXTIEN4	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
3	EXTIEN3	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
2	EXTIEN2	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
1	EXTIEN1	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			
0	EXTIEN0	0x0	RW	**External Pin Enable**
	External Pin interrupt enable			

25.6.36 GPIO_EM4WUEN - EM4 Wakeup Enable

Offset	Bit Position																													
0x42C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					EM4WUEN																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:16	EM4WUEN	0x0	RW	**EM4 wake up enable**
	Write 1 to enable EM4 wake up request, write 0 to disable EM4 wake up request			
15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

25.6.37 GPIO_EM4WUPOL - EM4 Wakeup Polarity

Offset	Bit Position																													
0x430	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset					0x0																											
Access					RW																											
Name					EM4WUPOL																											

Bit	Name	Reset	Access	Description
31:28	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
27:16	EM4WUPOL EM4 Wakeup Polarity	0x0	RW	**EM4 Wake-Up Polarity**
15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

25.6.38 GPIO_DBGROUTEPEN - Debugger Route Pin Enable

Offset	Bit Position																															
0x440	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x1	0x1	0x1	0x1
Access																													RW	RW	RW	RW
Name																													TDIPEN	TDOPEN	SWDIOTMSPEN	SWCLKTCKPEN

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	TDIPEN	0x1	RW	**JTAG Test Debug Input Pin Enable**
	Enable JTAG TDI connection to pin.			
2	TDOPEN	0x1	RW	**JTAG Test Debug Output Pin Enable**
	Enable JTAG TDO connection to pin.			
1	SWDIOTMSPEN	0x1	RW	**Route Location 0**
	Enable Serial Wire Data and JTAG Test Mode Select connection to pin. WARNING: When the pin is disabled, the device can no longer be accessed by a debugger. A reset will set the pin back to a default state as enabled. If you disable this pin, make sure you have at least a 3 second timeout at the start of your program code before you disable the pin. This way, the debugger will have time to halt the device after a reset before the pin is disabled.			
0	SWCLKTCKPEN	0x1	RW	**Route Pin Enable**
	Enable Serial Wire and JTAG CLock connection to pin. WARNING: When the pin is disabled, the device can no longer be accessed by a debugger. A reset will set the pin back to a default state as enabled. If you disable this pin, make sure you have at least a 3 second timeout at the start of your program code before you disable the pin. This way, the debugger will have time to halt the device after a reset before the pin is disabled.			

25.6.39 GPIO_TRACEROUTEPEN - Trace Route Pin Enable

Offset	Bit Position																													
0x444	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											TRACEDATA3PEN	TRACEDATA2PEN	TRACEDATA1PEN	TRACEDATA0PEN	TRACECLKPEN	SWVPEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	TRACEDATA3PEN Trace Data3 Pin Enable	0x0	RW	**Trace Data3 Pin Enable**
4	TRACEDATA2PEN Trace Data2 Pin Enable	0x0	RW	**Trace Data2 Pin Enable**
3	TRACEDATA1PEN Trace Data1 Pin Enable	0x0	RW	**Trace Data1 Pin Enable**
2	TRACEDATA0PEN Trace Data0 Pin Enable	0x0	RW	**Trace Data0 Pin Enable**
1	TRACECLKPEN Trace Clk Pin Enable	0x0	RW	**Trace Clk Pin Enable**
0	SWVPEN Serial Wire Viewer Output Pin Enable	0x0	RW	**Serial Wire Viewer Output Pin Enable**

25.6.40 GPIO_ACMP0_ROUTEEN - ACMP0 Pin Enable

Offset	Bit Position																													
0x450	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																ACMPOUTPEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	ACMPOUTPEN ACMPOUT pin enable control bit	0x0	RW	**ACMPOUT pin enable control bit**

25.6.41 GPIO_ACMP0_ACMPOUTROUTE - ACMPOUT Port/Pin Select

Offset	Bit Position																													
0x454	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ACMPOUT pin select register**
	ACMPOUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ACMPOUT port select register**
	ACMPOUT port select register			

25.6.42 GPIO_ACMP1_ROUTEEN - ACMP1 Pin Enable

Offset	Bit Position																													
0x45C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																ACMPOUTPEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	ACMPOUTPEN	0x0	RW	**ACMPOUT pin enable control bit**
	ACMPOUT pin enable control bit			

25.6.43 GPIO_ACMP1_ACMPOUTROUTE - ACMPOUT Port/Pin Select

Offset	Bit Position																													
0x460	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ACMPOUT pin select register**
	ACMPOUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ACMPOUT port select register**
	ACMPOUT port select register			

25.6.44 GPIO_EUSART0_ROUTEEN - EUSART0 Pin Enable

Offset	Bit Position																													
0x468	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												TXPEN	SCLKPEN	RXPEN	RTSPEN	CSPEN

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	TXPEN	0x0	RW	**TX pin enable control bit**
	TX pin enable control bit			
3	SCLKPEN	0x0	RW	**SCLK pin enable control bit**
	SCLK pin enable control bit			
2	RXPEN	0x0	RW	**RX pin enable control bit**
	RX pin enable control bit			
1	RTSPEN	0x0	RW	**RTS pin enable control bit**
	RTS pin enable control bit			
0	CSPEN	0x0	RW	**CS pin enable control bit**
	CS pin enable control bit			

25.6.45 GPIO_EUSART0_CSROUTE - CS Port/Pin Select

Offset	Bit Position																													
0x46C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CS pin select register	0x0	RW	**CS pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CS port select register	0x0	RW	**CS port select register**

25.6.46 GPIO_EUSART0_CTSROUTE - CTS Port/Pin Select

Offset	Bit Position																													
0x470	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CTS pin select register	0x0	RW	**CTS pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CTS port select register	0x0	RW	**CTS port select register**

25.6.47 GPIO_EUSART0_RTSROUTE - RTS Port/Pin Select

Offset	Bit Position																													
0x474	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN RTS pin select register	0x0	RW	**RTS pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT RTS port select register	0x0	RW	**RTS port select register**

25.6.48 GPIO_EUSART0_RXROUTE - RX Port/Pin Select

Offset	Bit Position																													
0x478	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN RX pin select register	0x0	RW	**RX pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT RX port select register	0x0	RW	**RX port select register**

25.6.49 GPIO_EUSART0_SCLKROUTE - SCLK Port/Pin Select

Offset	Bit Position																													
0x47C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN SCLK pin select register	0x0	RW	**SCLK pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT SCLK port select register	0x0	RW	**SCLK port select register**

25.6.50 GPIO_EUSART0_TXROUTE - TX Port/Pin Select

Offset	Bit Position																													
0x480	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN TX pin select register	0x0	RW	**TX pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT TX port select register	0x0	RW	**TX port select register**

25.6.51 GPIO_EUSART1_ROUTEEN - EUSART1 Pin Enable

Offset	Bit Position																													
0x488	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												TXPEN	SCLKPEN	RXPEN	RTSPEN	CSPEN

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	TXPEN	0x0	RW	**TX pin enable control bit**
	TX pin enable control bit			
3	SCLKPEN	0x0	RW	**SCLK pin enable control bit**
	SCLK pin enable control bit			
2	RXPEN	0x0	RW	**RX pin enable control bit**
	RX pin enable control bit			
1	RTSPEN	0x0	RW	**RTS pin enable control bit**
	RTS pin enable control bit			
0	CSPEN	0x0	RW	**CS pin enable control bit**
	CS pin enable control bit			

25.6.52 GPIO_EUSART1_CSROUTE - CS Port/Pin Select

Offset	Bit Position																													
0x48C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CS pin select register**
	CS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CS port select register**
	CS port select register			

25.6.53 GPIO_EUSART1_CTSROUTE - CTS Port/Pin Select

Offset	Bit Position																													
0x490	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CTS pin select register	0x0	RW	**CTS pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CTS port select register	0x0	RW	**CTS port select register**

25.6.54 GPIO_EUSART1_RTSROUTE - RTS Port/Pin Select

Offset	Bit Position																													
0x494	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN RTS pin select register	0x0	RW	**RTS pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT RTS port select register	0x0	RW	**RTS port select register**

25.6.55 GPIO_EUSART1_RXROUTE - RX Port/Pin Select

Offset	Bit Position																													
0x498	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN RX pin select register	0x0	RW	**RX pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT RX port select register	0x0	RW	**RX port select register**

25.6.56 GPIO_EUSART1_SCLKROUTE - SCLK Port/Pin Select

Offset	Bit Position																													
0x49C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN SCLK pin select register	0x0	RW	**SCLK pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT SCLK port select register	0x0	RW	**SCLK port select register**

25.6.57 GPIO_EUSART1_TXROUTE - TX Port/Pin Select

Offset	Bit Position																													
0x4A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN TX pin select register	0x0	RW	**TX pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT TX port select register	0x0	RW	**TX port select register**

25.6.58 GPIO_EUSART2_ROUTEEN - EUSART2 Pin Enable

Offset	Bit Position																													
0x4A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												TXPEN	SCLKPEN	RXPEN	RTSPEN	CSPEN

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	TXPEN TX pin enable control bit	0x0	RW	**TX pin enable control bit**
3	SCLKPEN SCLK pin enable control bit	0x0	RW	**SCLK pin enable control bit**
2	RXPEN RX pin enable control bit	0x0	RW	**RX pin enable control bit**
1	RTSPEN RTS pin enable control bit	0x0	RW	**RTS pin enable control bit**
0	CSPEN CS pin enable control bit	0x0	RW	**CS pin enable control bit**

25.6.59 GPIO_EUSART2_CSROUTE - CS Port/Pin Select

Offset	Bit Position																													
0x4AC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CS pin select register	0x0	RW	**CS pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CS port select register	0x0	RW	**CS port select register**

25.6.60 GPIO_EUSART2_CTSROUTE - CTS Port/Pin Select

Offset	Bit Position																													
0x4B0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CTS pin select register	0x0	RW	**CTS pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CTS port select register	0x0	RW	**CTS port select register**

25.6.61 GPIO_EUSART2_RTSROUTE - RTS Port/Pin Select

Offset	Bit Position																													
0x4B4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN RTS pin select register	0x0	RW	**RTS pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT RTS port select register	0x0	RW	**RTS port select register**

25.6.62 GPIO_EUSART2_RXROUTE - RX Port/Pin Select

Offset	Bit Position																													
0x4B8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN RX pin select register	0x0	RW	**RX pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT RX port select register	0x0	RW	**RX port select register**

25.6.63 GPIO_EUSART2_SCLKROUTE - SCLK Port/Pin Select

Offset	Bit Position																													
0x4BC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN SCLK pin select register	0x0	RW	**SCLK pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT SCLK port select register	0x0	RW	**SCLK port select register**

25.6.64 GPIO_EUSART2_TXROUTE - TX Port/Pin Select

Offset	Bit Position																													
0x4C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN TX pin select register	0x0	RW	**TX pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT TX port select register	0x0	RW	**TX port select register**

25.6.65 GPIO_FRC_ROUTEEN - FRC Pin Enable

Offset	Bit Position																													
0x4C8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														DOUTPEN	DFRAMEPEN	DCLKPEN

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	DOUTPEN	0x0	RW	**DOUT pin enable control bit**
	DOUT pin enable control bit			
1	DFRAMEPEN	0x0	RW	**DFRAME pin enable control bit**
	DFRAME pin enable control bit			
0	DCLKPEN	0x0	RW	**DCLK pin enable control bit**
	DCLK pin enable control bit			

25.6.66 GPIO_FRC_DCLKROUTE - DCLK Port/Pin Select

Offset	Bit Position																													
0x4CC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**DCLK pin select register**
	DCLK pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**DCLK port select register**
	DCLK port select register			

25.6.67 GPIO_FRC_DFRAMEROUTE - DFRAME Port/Pin Select

Offset	Bit Position																													
0x4D0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**DFRAME pin select register**
	DFRAME pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**DFRAME port select register**
	DFRAME port select register			

25.6.68 GPIO_FRC_DOUTROUTE - DOUT Port/Pin Select

Offset	Bit Position																													
0x4D4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**DOUT pin select register**
	DOUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**DOUT port select register**
	DOUT port select register			

25.6.69 GPIO_CMU_ROUTEEN - HOSTCMU Pin Enable

Offset	Bit Position																													
0x4DC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														CLKOUT2PEN	CLKOUT1PEN	CLKOUT0PEN

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	CLKOUT2PEN	0x0	RW	**CLKOUT2 pin enable control bit**
	CLKOUT2 pin enable control bit			
1	CLKOUT1PEN	0x0	RW	**CLKOUT1 pin enable control bit**
	CLKOUT1 pin enable control bit			
0	CLKOUT0PEN	0x0	RW	**CLKOUT0 pin enable control bit**
	CLKOUT0 pin enable control bit			

25.6.70 GPIO_CMU_CLKIN0ROUTE - CLKIN0 Port/Pin Select

Offset	Bit Position																													
0x4E0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CLKIN0 pin select register**
	CLKIN0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CLKIN0 port select register**
	CLKIN0 port select register			

25.6.71 GPIO_CMU_CLKOUT0ROUTE - CLKOUT0 Port/Pin Select

Offset	Bit Position																													
0x4E4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CLKOUT0 pin select register**
	CLKOUT0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CLKOUT0 port select register**
	CLKOUT0 port select register			

25.6.72 GPIO_CMU_CLKOUT1ROUTE - CLKOUT1 Port/Pin Select

Offset	Bit Position																													
0x4E8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CLKOUT1 pin select register**
	CLKOUT1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CLKOUT1 port select register**
	CLKOUT1 port select register			

25.6.73 GPIO_CMU_CLKOUT2ROUTE - CLKOUT2 Port/Pin Select

Offset	Bit Position																													
0x4EC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CLKOUT2 pin select register**
	CLKOUT2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CLKOUT2 port select register**
	CLKOUT2 port select register			

25.6.74 GPIO_I2C0_ROUTEEN - I2C0 Pin Enable

Offset	Bit Position																													
0x4F8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															SDAPEN	SCLPEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	SDAPEN	0x0	RW	**SDA pin enable control bit**
	SDA pin enable control bit			
0	SCLPEN	0x0	RW	**SCL pin enable control bit**
	SCL pin enable control bit			

25.6.75 GPIO_I2C0_SCLROUTE - SCL Port/Pin Select

Offset	Bit Position																													
0x4FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN SCL pin select register	0x0	RW	**SCL pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT SCL port select register	0x0	RW	**SCL port select register**

25.6.76 GPIO_I2C0_SDAROUTE - SDA Port/Pin Select

Offset	Bit Position																													
0x500	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN SDA pin select register	0x0	RW	**SDA pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT SDA port select register	0x0	RW	**SDA port select register**

25.6.77 GPIO_I2C1_ROUTEEN - I2C1 Pin Enable

Offset	Bit Position																													
0x508	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															SDAPEN	SCLPEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	SDAPEN SDA pin enable control bit	0x0	RW	**SDA pin enable control bit**
0	SCLPEN SCL pin enable control bit	0x0	RW	**SCL pin enable control bit**

25.6.78 GPIO_I2C1_SCLROUTE - SCL Port/Pin Select

Offset	Bit Position																													
0x50C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN SCL pin select register	0x0	RW	**SCL pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT SCL port select register	0x0	RW	**SCL port select register**

25.6.79 GPIO_I2C1_SDAROUTE - SDA Port/Pin Select

Offset	Bit Position																													
0x510	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN SDA pin select register	0x0	RW	**SDA pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT SDA port select register	0x0	RW	**SDA port select register**

25.6.80 GPIO_I2C2_ROUTEEN - I2C2 Pin Enable

Offset	Bit Position																													
0x518	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															SDAPEN	SCLPEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	SDAPEN SDA pin enable control bit	0x0	RW	**SDA pin enable control bit**
0	SCLPEN SCL pin enable control bit	0x0	RW	**SCL pin enable control bit**

25.6.81 GPIO_I2C2_SCLROUTE - SCL Port/Pin Select

Offset	Bit Position																													
0x51C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN SCL pin select register	0x0	RW	**SCL pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT SCL port select register	0x0	RW	**SCL port select register**

25.6.82 GPIO_I2C2_SDAROUTE - SDA Port/Pin Select

Offset	Bit Position																													
0x520	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN SDA pin select register	0x0	RW	**SDA pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT SDA port select register	0x0	RW	**SDA port select register**

25.6.83 GPIO_LETIMER0_ROUTEEN - LETIMER0 Pin Enable

Offset	Bit Position																													
0x528	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															OUT1PEN	OUT0PEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	OUT1PEN	0x0	RW	**OUT1 pin enable control bit**
	OUT1 pin enable control bit			
0	OUT0PEN	0x0	RW	**OUT0 pin enable control bit**
	OUT0 pin enable control bit			

25.6.84 GPIO_LETIMER0_OUT0ROUTE - OUT0 Port/Pin Select

Offset	Bit Position																													
0x52C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**OUT0 pin select register**
	OUT0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**OUT0 port select register**
	OUT0 port select register			

25.6.85 GPIO_LETIMER0_OUT1ROUTE - OUT1 Port/Pin Select

Offset	Bit Position																													
0x530	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**OUT1 pin select register**
	OUT1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**OUT1 port select register**
	OUT1 port select register			

25.6.86 GPIO_LFXO_ROUTEEN - LFXO Pin Enable

Offset	Bit Position																													
0x538	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																LFXOCLKLVRAWPEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	LFXOCLKLVRAWPEN	0x0	RW	**LFXOCLKLVRAW pin enable control bit**
	LFXOCLKLVRAW pin enable control bit			

25.6.87 GPIO_LFXO_LFXOCLKLVRAWROUTE - LFXOCLKLVRAW Port/Pin Select

Offset	Bit Position																													
0x53C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**LFXOCLKLVRAW pin select register**
	LFXOCLKLVRAW pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**LFXOCLKLVRAW port select register**
	LFXOCLKLVRAW port select register			

25.6.88 GPIO_MODEM_ROUTEEN - MODEM Pin Enable

Offset	Bit Position																													
0x544	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																		0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																		DOUTPEN	DCLKPEN	ANTTRIGSTOPPEN	ANTTRIGPEN	ANTSWUSPEN	ANTSWENPEN	ANTRR5PEN	ANTRR4PEN	ANTRR3PEN	ANTRR2PEN	ANTRR1PEN	ANTRR0PEN	ANTROLLOVERPEN	ANT1PEN	ANT0PEN

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14	DOUTPEN	0x0	RW	**DOUT pin enable control bit**
	DOUT pin enable control bit			
13	DCLKPEN	0x0	RW	**DCLK pin enable control bit**
	DCLK pin enable control bit			
12	ANTTRIGSTOPPEN	0x0	RW	**ANTTRIGSTOP pin enable control bit**
	ANTTRIGSTOP pin enable control bit			
11	ANTTRIGPEN	0x0	RW	**ANTTRIG pin enable control bit**
	ANTTRIG pin enable control bit			
10	ANTSWUSPEN	0x0	RW	**ANTSWUS pin enable control bit**
	ANTSWUS pin enable control bit			
9	ANTSWENPEN	0x0	RW	**ANTSWEN pin enable control bit**
	ANTSWEN pin enable control bit			
8	ANTRR5PEN	0x0	RW	**ANTRR5 pin enable control bit**
	ANTRR5 pin enable control bit			
7	ANTRR4PEN	0x0	RW	**ANTRR4 pin enable control bit**
	ANTRR4 pin enable control bit			
6	ANTRR3PEN	0x0	RW	**ANTRR3 pin enable control bit**
	ANTRR3 pin enable control bit			
5	ANTRR2PEN	0x0	RW	**ANTRR2 pin enable control bit**
	ANTRR2 pin enable control bit			
4	ANTRR1PEN	0x0	RW	**ANTRR1 pin enable control bit**
	ANTRR1 pin enable control bit			
3	ANTRR0PEN	0x0	RW	**ANTRR0 pin enable control bit**
	ANTRR0 pin enable control bit			
2	ANTROLLOVERPEN	0x0	RW	**ANTROLLOVER pin enable control bit**
	ANTROLLOVER pin enable control bit			
1	ANT1PEN	0x0	RW	**ANT1 pin enable control bit**
	ANT1 pin enable control bit			

Bit	Name	Reset	Access	Description
0	ANT0PEN	0x0	RW	**ANT0 pin enable control bit**
	ANT0 pin enable control bit			

25.6.89 GPIO_MODEM_ANT0ROUTE - ANT0 Port/Pin Select

Offset	Bit Position																													
0x548	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANT0 pin select register**
	ANT0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANT0 port select register**
	ANT0 port select register			

25.6.90 GPIO_MODEM_ANT1ROUTE - ANT1 Port/Pin Select

Offset	Bit Position																													
0x54C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANT1 pin select register**
	ANT1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANT1 port select register**
	ANT1 port select register			

25.6.91 GPIO_MODEM_ANTROLLOVERROUTE - ANTROLLOVER Port/Pin Select

Offset	Bit Position																													
0x550	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTROLLOVER pin select register**
	ANTROLLOVER pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTROLLOVER port select register**
	ANTROLLOVER port select register			

25.6.92 GPIO_MODEM_ANTRR0ROUTE - ANTRR0 Port/Pin Select

Offset	Bit Position																													
0x554	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTRR0 pin select register**
	ANTRR0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTRR0 port select register**
	ANTRR0 port select register			

25.6.93 GPIO_MODEM_ANTRR1ROUTE - ANTRR1 Port/Pin Select

Offset	Bit Position																													
0x558	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTRR1 pin select register**
	ANTRR1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTRR1 port select register**
	ANTRR1 port select register			

25.6.94 GPIO_MODEM_ANTRR2ROUTE - ANTRR2 Port/Pin Select

Offset	Bit Position																													
0x55C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTRR2 pin select register**
	ANTRR2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTRR2 port select register**
	ANTRR2 port select register			

25.6.95 GPIO_MODEM_ANTRR3ROUTE - ANTRR3 Port/Pin Select

Offset	Bit Position																													
0x560	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTRR3 pin select register**
	ANTRR3 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTRR3 port select register**
	ANTRR3 port select register			

25.6.96 GPIO_MODEM_ANTRR4ROUTE - ANTRR4 Port/Pin Select

Offset	Bit Position																													
0x564	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTRR4 pin select register**
	ANTRR4 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTRR4 port select register**
	ANTRR4 port select register			

25.6.97 GPIO_MODEM_ANTRR5ROUTE - ANTRR5 Port/Pin Select

Offset	Bit Position																													
0x568	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTRR5 pin select register**
	ANTRR5 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTRR5 port select register**
	ANTRR5 port select register			

25.6.98 GPIO_MODEM_ANTSWENROUTE - ANTSWEN Port/Pin Select

Offset	Bit Position																													
0x56C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTSWEN pin select register**
	ANTSWEN pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTSWEN port select register**
	ANTSWEN port select register			

25.6.99 GPIO_MODEM_ANTSWUSROUTE - ANTSWUS Port/Pin Select

Offset	Bit Position																													
0x570	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTSWUS pin select register**
	ANTSWUS pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTSWUS port select register**
	ANTSWUS port select register			

25.6.100 GPIO_MODEM_ANTTRIGROUTE - ANTTRIG Port/Pin Select

Offset	Bit Position																													
0x574	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTTRIG pin select register**
	ANTTRIG pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTTRIG port select register**
	ANTTRIG port select register			

25.6.101 GPIO_MODEM_ANTTRIGSTOPROUTE - ANTTRIGSTOP Port/Pin Select

Offset	Bit Position																													
0x578	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ANTTRIGSTOP pin select register**
	ANTTRIGSTOP pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ANTTRIGSTOP port select register**
	ANTTRIGSTOP port select register			

25.6.102 GPIO_MODEM_DCLKROUTE - DCLK Port/Pin Select

Offset	Bit Position																													
0x57C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**DCLK pin select register**
	DCLK pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**DCLK port select register**
	DCLK port select register			

25.6.103 GPIO_MODEM_DINROUTE - DIN Port/Pin Select

Offset	Bit Position																													
0x580	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN DIN pin select register	0x0	RW	**DIN pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT DIN port select register	0x0	RW	**DIN port select register**

25.6.104 GPIO_MODEM_DOUTROUTE - DOUT Port/Pin Select

Offset	Bit Position																													
0x584	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN DOUT pin select register	0x0	RW	**DOUT pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT DOUT port select register	0x0	RW	**DOUT port select register**

25.6.105 GPIO_PCNT0_S0INROUTE - S0IN Port/Pin Select

Offset	Bit Position																													
0x590	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN S0IN pin select register	0x0	RW	**S0IN pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT S0IN port select register	0x0	RW	**S0IN port select register**

25.6.106 GPIO_PCNT0_S1INROUTE - S1IN Port/Pin Select

Offset	Bit Position																													
0x594	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN S1IN pin select register	0x0	RW	**S1IN pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT S1IN port select register	0x0	RW	**S1IN port select register**

25.6.107 GPIO_PIXELRZ0_ROUTEEN - PIXELRZ0 Pin Enable

Offset	Bit Position																															
0x59C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																RZTXOUTPEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	RZTXOUTPEN	0x0	RW	**RZTXOUT pin enable control bit**
	RZTXOUT pin enable control bit			

25.6.108 GPIO_PIXELRZ0_RZTXOUTROUTE - RZTXOUT Port/Pin Select

Offset	Bit Position																															
0x5A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**RZTXOUT pin select register**
	RZTXOUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**RZTXOUT port select register**
	RZTXOUT port select register			

25.6.109 GPIO_PIXELRZ1_ROUTEEN - PIXELRZ1 Pin Enable

Offset	Bit Position																															
0x5A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																RW
Name																																RZTXOUTPEN

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	RZTXOUTPEN	0x0	RW	**RZTXOUT pin enable control bit**
	RZTXOUT pin enable control bit			

25.6.110 GPIO_PIXELRZ1_RZTXOUTROUTE - RZTXOUT Port/Pin Select

Offset	Bit Position																															
0x5AC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**RZTXOUT pin select register**
	RZTXOUT pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**RZTXOUT port select register**
	RZTXOUT port select register			

25.6.111 GPIO_PRS0_ROUTEEN - PRS0 Pin Enable

Offset	Bit Position																													
0x5B4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																	SYNCH3PEN	SYNCH2PEN	SYNCH1PEN	SYNCH0PEN	ASYNCH11PEN	ASYNCH10PEN	ASYNCH9PEN	ASYNCH8PEN	ASYNCH7PEN	ASYNCH6PEN	ASYNCH5PEN	ASYNCH4PEN	ASYNCH3PEN	ASYNCH2PEN	ASYNCH1PEN	ASYNCH0PEN

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15	SYNCH3PEN	0x0	RW	**SYNCH3 pin enable control bit**
	SYNCH3 pin enable control bit			
14	SYNCH2PEN	0x0	RW	**SYNCH2 pin enable control bit**
	SYNCH2 pin enable control bit			
13	SYNCH1PEN	0x0	RW	**SYNCH1 pin enable control bit**
	SYNCH1 pin enable control bit			
12	SYNCH0PEN	0x0	RW	**SYNCH0 pin enable control bit**
	SYNCH0 pin enable control bit			
11	ASYNCH11PEN	0x0	RW	**ASYNCH11 pin enable control bit**
	ASYNCH11 pin enable control bit			
10	ASYNCH10PEN	0x0	RW	**ASYNCH10 pin enable control bit**
	ASYNCH10 pin enable control bit			
9	ASYNCH9PEN	0x0	RW	**ASYNCH9 pin enable control bit**
	ASYNCH9 pin enable control bit			
8	ASYNCH8PEN	0x0	RW	**ASYNCH8 pin enable control bit**
	ASYNCH8 pin enable control bit			
7	ASYNCH7PEN	0x0	RW	**ASYNCH7 pin enable control bit**
	ASYNCH7 pin enable control bit			
6	ASYNCH6PEN	0x0	RW	**ASYNCH6 pin enable control bit**
	ASYNCH6 pin enable control bit			
5	ASYNCH5PEN	0x0	RW	**ASYNCH5 pin enable control bit**
	ASYNCH5 pin enable control bit			
4	ASYNCH4PEN	0x0	RW	**ASYNCH4 pin enable control bit**
	ASYNCH4 pin enable control bit			
3	ASYNCH3PEN	0x0	RW	**ASYNCH3 pin enable control bit**
	ASYNCH3 pin enable control bit			
2	ASYNCH2PEN	0x0	RW	**ASYNCH2 pin enable control bit**
	ASYNCH2 pin enable control bit			
1	ASYNCH1PEN	0x0	RW	**ASYNCH1 pin enable control bit**

Bit	Name	Reset	Access	Description
	ASYNCH1 pin enable control bit			
0	ASYNCH0PEN	0x0	RW	**ASYNCH0 pin enable control bit**
	ASYNCH0 pin enable control bit			

25.6.112 GPIO_PRS0_ASYNCH0ROUTE - ASYNCH0 Port/Pin Select

Offset	Bit Position																													
0x5B8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH0 pin select register**
	ASYNCH0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH0 port select register**
	ASYNCH0 port select register			

25.6.113 GPIO_PRS0_ASYNCH1ROUTE - ASYNCH1 Port/Pin Select

Offset	Bit Position																													
0x5BC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH1 pin select register**
	ASYNCH1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH1 port select register**
	ASYNCH1 port select register			

25.6.114 GPIO_PRS0_ASYNCH2ROUTE - ASYNCH2 Port/Pin Select

Offset	Bit Position																													
0x5C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH2 pin select register**
	ASYNCH2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH2 port select register**
	ASYNCH2 port select register			

25.6.115 GPIO_PRS0_ASYNCH3ROUTE - ASYNCH3 Port/Pin Select

Offset	Bit Position																													
0x5C4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH3 pin select register**
	ASYNCH3 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH3 port select register**
	ASYNCH3 port select register			

25.6.116 GPIO_PRS0_ASYNCH4ROUTE - ASYNCH4 Port/Pin Select

Offset	Bit Position																													
0x5C8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH4 pin select register**
	ASYNCH4 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH4 port select register**
	ASYNCH4 port select register			

25.6.117 GPIO_PRS0_ASYNCH5ROUTE - ASYNCH5 Port/Pin Select

Offset	Bit Position																													
0x5CC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH5 pin select register**
	ASYNCH5 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH5 port select register**
	ASYNCH5 port select register			

25.6.118 GPIO_PRS0_ASYNCH6ROUTE - ASYNCH6 Port/Pin Select

Offset	Bit Position																													
0x5D0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH6 pin select register**
	ASYNCH6 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH6 port select register**
	ASYNCH6 port select register			

25.6.119 GPIO_PRS0_ASYNCH7ROUTE - ASYNCH7 Port/Pin Select

Offset	Bit Position																													
0x5D4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH7 pin select register**
	ASYNCH7 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH7 port select register**
	ASYNCH7 port select register			

25.6.120 GPIO_PRS0_ASYNCH8ROUTE - ASYNCH8 Port/Pin Select

Offset	Bit Position																													
0x5D8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH8 pin select register**
	ASYNCH8 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH8 port select register**
	ASYNCH8 port select register			

25.6.121 GPIO_PRS0_ASYNCH9ROUTE - ASYNCH9 Port/Pin Select

Offset	Bit Position																													
0x5DC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH9 pin select register**
	ASYNCH9 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH9 port select register**
	ASYNCH9 port select register			

25.6.122 GPIO_PRS0_ASYNCH10ROUTE - ASYNCH10 Port/Pin Select

Offset	Bit Position																													
0x5E0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH10 pin select register**
	ASYNCH10 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH10 port select register**
	ASYNCH10 port select register			

25.6.123 GPIO_PRS0_ASYNCH11ROUTE - ASYNCH11 Port/Pin Select

Offset	Bit Position																													
0x5E4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**ASYNCH11 pin select register**
	ASYNCH11 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**ASYNCH11 port select register**
	ASYNCH11 port select register			

25.6.124 GPIO_PRS0_SYNCH0ROUTE - SYNCH0 Port/Pin Select

Offset	Bit Position																													
0x5E8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SYNCH0 pin select register**
	SYNCH0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SYNCH0 port select register**
	SYNCH0 port select register			

25.6.125 GPIO_PRS0_SYNCH1ROUTE - SYNCH1 Port/Pin Select

Offset	Bit Position																													
0x5EC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SYNCH1 pin select register**
	SYNCH1 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SYNCH1 port select register**
	SYNCH1 port select register			

25.6.126 GPIO_PRS0_SYNCH2ROUTE - SYNCH2 Port/Pin Select

Offset	Bit Position																													
0x5F0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SYNCH2 pin select register**
	SYNCH2 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SYNCH2 port select register**
	SYNCH2 port select register			

25.6.127 GPIO_PRS0_SYNCH3ROUTE - SYNCH3 Port/Pin Select

Offset	Bit Position																													
0x5F4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**SYNCH3 pin select register**
	SYNCH3 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**SYNCH3 port select register**
	SYNCH3 port select register			

25.6.128 GPIO_RAC_ROUTEEN - RAC Pin Enable

Offset	Bit Position																													
0x5FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															RW	RW
Name																															PAENPEN	LNAENPEN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	PAENPEN	0x0	RW	**PAEN pin enable control bit**
	PAEN pin enable control bit			
0	LNAENPEN	0x0	RW	**LNAEN pin enable control bit**
	LNAEN pin enable control bit			

25.6.129 GPIO_RAC_LNAENROUTE - LNAEN Port/Pin Select

Offset	Bit Position																													
0x600	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**LNAEN pin select register**
	LNAEN pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**LNAEN port select register**
	LNAEN port select register			

25.6.130 GPIO_RAC_PAENROUTE - PAEN Port/Pin Select

Offset	Bit Position																													
0x604	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN PAEN pin select register	0x0	RW	**PAEN pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT PAEN port select register	0x0	RW	**PAEN port select register**

25.6.131 GPIO_TIMER0_ROUTEEN - TIMER0 Pin Enable

Offset	Bit Position																													
0x668	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											CCC2PEN	CCC1PEN	CCC0PEN	CC2PEN	CC1PEN	CC0PEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CCC2PEN CDTI2 pin enable control bit	0x0	RW	**CDTI2 pin enable control bit**
4	CCC1PEN CDTI1 pin enable control bit	0x0	RW	**CDTI1 pin enable control bit**
3	CCC0PEN CDTI0 pin enable control bit	0x0	RW	**CDTI0 pin enable control bit**
2	CC2PEN CC2 pin enable control bit	0x0	RW	**CC2 pin enable control bit**
1	CC1PEN CC1 pin enable control bit	0x0	RW	**CC1 pin enable control bit**
0	CC0PEN CC0 pin enable control bit	0x0	RW	**CC0 pin enable control bit**

25.6.132 GPIO_TIMER0_CC0ROUTE - CC0 Port/Pin Select

Offset	Bit Position																													
0x66C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC0 pin select register	0x0	RW	**CC0 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC0 port select register	0x0	RW	**CC0 port select register**

25.6.133 GPIO_TIMER0_CC1ROUTE - CC1 Port/Pin Select

Offset	Bit Position																													
0x670	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC1 pin select register	0x0	RW	**CC1 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC1 port select register	0x0	RW	**CC1 port select register**

25.6.134 GPIO_TIMER0_CC2ROUTE - CC2 Port/Pin Select

Offset	Bit Position																													
0x674	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC2 pin select register	0x0	RW	**CC2 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC2 port select register	0x0	RW	**CC2 port select register**

25.6.135 GPIO_TIMER0_CDTI0ROUTE - CDTI0 Port/Pin Select

Offset	Bit Position																													
0x678	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI0 pin select register	0x0	RW	**CDTI0 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI0 port select register	0x0	RW	**CDTI0 port select register**

25.6.136 GPIO_TIMER0_CDTI1ROUTE - CDTI1 Port/Pin Select

Offset	Bit Position																													
0x67C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI1 pin select register	0x0	RW	**CDTI1 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI1 port select register	0x0	RW	**CDTI1 port select register**

25.6.137 GPIO_TIMER0_CDTI2ROUTE - CDTI2 Port/Pin Select

Offset	Bit Position																													
0x680	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI2 pin select register	0x0	RW	**CDTI2 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI2 port select register	0x0	RW	**CDTI2 port select register**

25.6.138 GPIO_TIMER1_ROUTEEN - TIMER1 Pin Enable

Offset	Bit Position																													
0x688	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																											0x0	0x0	0x0	0x0	0x0	0x0
Access																											RW	RW	RW	RW	RW	RW
Name																											CCC2PEN	CCC1PEN	CCC0PEN	CC2PEN	CC1PEN	CC0PEN

Bit	Name	Reset	Access	Description
31:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	CCC2PEN	0x0	RW	**CDTI2 pin enable control bit**
	CDTI2 pin enable control bit			
4	CCC1PEN	0x0	RW	**CDTI1 pin enable control bit**
	CDTI1 pin enable control bit			
3	CCC0PEN	0x0	RW	**CDTI0 pin enable control bit**
	CDTI0 pin enable control bit			
2	CC2PEN	0x0	RW	**CC2 pin enable control bit**
	CC2 pin enable control bit			
1	CC1PEN	0x0	RW	**CC1 pin enable control bit**
	CC1 pin enable control bit			
0	CC0PEN	0x0	RW	**CC0 pin enable control bit**
	CC0 pin enable control bit			

25.6.139 GPIO_TIMER1_CC0ROUTE - CC0 Port/Pin Select

Offset	Bit Position																													
0x68C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN	0x0	RW	**CC0 pin select register**
	CC0 pin select register			
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT	0x0	RW	**CC0 port select register**
	CC0 port select register			

25.6.140 GPIO_TIMER1_CC1ROUTE - CC1 Port/Pin Select

Offset	Bit Position																													
0x690	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC1 pin select register	0x0	RW	**CC1 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC1 port select register	0x0	RW	**CC1 port select register**

25.6.141 GPIO_TIMER1_CC2ROUTE - CC2 Port/Pin Select

Offset	Bit Position																													
0x694	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC2 pin select register	0x0	RW	**CC2 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC2 port select register	0x0	RW	**CC2 port select register**

25.6.142 GPIO_TIMER1_CDTI0ROUTE - CDTI0 Port/Pin Select

Offset	Bit Position																													
0x698	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI0 pin select register	0x0	RW	**CDTI0 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI0 port select register	0x0	RW	**CDTI0 port select register**

25.6.143 GPIO_TIMER1_CDTI1ROUTE - CDTI1 Port/Pin Select

Offset	Bit Position																													
0x69C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI1 pin select register	0x0	RW	**CDTI1 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI1 port select register	0x0	RW	**CDTI1 port select register**

25.6.144 GPIO_TIMER1_CDTI2ROUTE - CDTI2 Port/Pin Select

Offset	Bit Position																													
0x6A0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI2 pin select register	0x0	RW	**CDTI2 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI2 port select register	0x0	RW	**CDTI2 port select register**

25.6.145 GPIO_TIMER2_ROUTEEN - TIMER2 Pin Enable

Offset	Bit Position																													
0x6A8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																			0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																			CCC6PEN	CCC5PEN	CCC4PEN	CCC3PEN	CCC2PEN	CCC1PEN	CCC0PEN	CC6PEN	CC5PEN	CC4PEN	CC3PEN	CC2PEN	CC1PEN	CC0PEN

Bit	Name	Reset	Access	Description
31:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	CCC6PEN	0x0	RW	**CDTI6 pin enable control bit**
	CDTI6 pin enable control bit			
12	CCC5PEN	0x0	RW	**CDTI5 pin enable control bit**
	CDTI5 pin enable control bit			
11	CCC4PEN	0x0	RW	**CDTI4 pin enable control bit**
	CDTI4 pin enable control bit			
10	CCC3PEN	0x0	RW	**CDTI3 pin enable control bit**
	CDTI3 pin enable control bit			
9	CCC2PEN	0x0	RW	**CDTI2 pin enable control bit**
	CDTI2 pin enable control bit			
8	CCC1PEN	0x0	RW	**CDTI1 pin enable control bit**
	CDTI1 pin enable control bit			
7	CCC0PEN	0x0	RW	**CDTI0 pin enable control bit**
	CDTI0 pin enable control bit			
6	CC6PEN	0x0	RW	**CC6 pin enable control bit**
	CC6 pin enable control bit			
5	CC5PEN	0x0	RW	**CC5 pin enable control bit**
	CC5 pin enable control bit			
4	CC4PEN	0x0	RW	**CC4 pin enable control bit**
	CC4 pin enable control bit			
3	CC3PEN	0x0	RW	**CC3 pin enable control bit**
	CC3 pin enable control bit			
2	CC2PEN	0x0	RW	**CC2 pin enable control bit**
	CC2 pin enable control bit			
1	CC1PEN	0x0	RW	**CC1 pin enable control bit**
	CC1 pin enable control bit			
0	CC0PEN	0x0	RW	**CC0 pin enable control bit**
	CC0 pin enable control bit			

25.6.146 GPIO_TIMER2_CC0ROUTE - CC0 Port/Pin Select

Offset	Bit Position																													
0x6AC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC0 pin select register	0x0	RW	**CC0 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC0 port select register	0x0	RW	**CC0 port select register**

25.6.147 GPIO_TIMER2_CC1ROUTE - CC1 Port/Pin Select

Offset	Bit Position																													
0x6B0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC1 pin select register	0x0	RW	**CC1 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC1 port select register	0x0	RW	**CC1 port select register**

25.6.148 GPIO_TIMER2_CC2ROUTE - CC2 Port/Pin Select

Offset	Bit Position																													
0x6B4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC2 pin select register	0x0	RW	**CC2 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC2 port select register	0x0	RW	**CC2 port select register**

25.6.149 GPIO_TIMER2_CC3ROUTE - CC3 Port/Pin Select

Offset	Bit Position																													
0x6B8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC3 pin select register	0x0	RW	**CC3 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC3 port select register	0x0	RW	**CC3 port select register**

25.6.150 GPIO_TIMER2_CC4ROUTE - CC4 Port/Pin Select

Offset	Bit Position																													
0x6BC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC4 pin select register	0x0	RW	**CC4 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC4 port select register	0x0	RW	**CC4 port select register**

25.6.151 GPIO_TIMER2_CC5ROUTE - CC5 Port/Pin Select

Offset	Bit Position																													
0x6C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC5 pin select register	0x0	RW	**CC5 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC5 port select register	0x0	RW	**CC5 port select register**

25.6.152 GPIO_TIMER2_CC6ROUTE - CC6 Port/Pin Select

Offset	Bit Position																													
0x6C4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC6 pin select register	0x0	RW	**CC6 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC6 port select register	0x0	RW	**CC6 port select register**

25.6.153 GPIO_TIMER2_CDTI0ROUTE - CDTI0 Port/Pin Select

Offset	Bit Position																													
0x6C8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI0 pin select register	0x0	RW	**CDTI0 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI0 port select register	0x0	RW	**CDTI0 port select register**

25.6.154 GPIO_TIMER2_CDTI1ROUTE - CDTI1 Port/Pin Select

Offset	Bit Position																													
0x6CC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI1 pin select register	0x0	RW	**CDTI1 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI1 port select register	0x0	RW	**CDTI1 port select register**

25.6.155 GPIO_TIMER2_CDTI2ROUTE - CDTI2 Port/Pin Select

Offset	Bit Position																													
0x6D0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI2 pin select register	0x0	RW	**CDTI2 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI2 port select register	0x0	RW	**CDTI2 port select register**

25.6.156 GPIO_TIMER2_CDTI3ROUTE - CDTI3 Port/Pin Select

Offset	Bit Position																													
0x6D4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI3 pin select register	0x0	RW	**CDTI3 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI3 port select register	0x0	RW	**CDTI3 port select register**

25.6.157 GPIO_TIMER2_CDTI4ROUTE - CDTI4 Port/Pin Select

Offset	Bit Position																													
0x6D8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI4 pin select register	0x0	RW	**CDTI4 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI4 port select register	0x0	RW	**CDTI4 port select register**

25.6.158 GPIO_TIMER2_CDTI5ROUTE - CDTI5 Port/Pin Select

Offset	Bit Position																													
0x6DC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI5 pin select register	0x0	RW	**CDTI5 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI5 port select register	0x0	RW	**CDTI5 port select register**

25.6.159 GPIO_TIMER2_CDTI6ROUTE - CDTI6 Port/Pin Select

Offset	Bit Position																													
0x6E0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI6 pin select register	0x0	RW	**CDTI6 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI6 port select register	0x0	RW	**CDTI6 port select register**

25.6.160 GPIO_TIMER3_ROUTEEN - TIMER3 Pin Enable

Offset	Bit Position																													
0x6E8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																			0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Name																			CCC6PEN	CCC5PEN	CCC4PEN	CCC3PEN	CCC2PEN	CCC1PEN	CCC0PEN	CC6PEN	CC5PEN	CC4PEN	CC3PEN	CC2PEN	CC1PEN	CC0PEN

Bit	Name	Reset	Access	Description
31:14	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
13	CCC6PEN	0x0	RW	**CDTI6 pin enable control bit**
	CDTI6 pin enable control bit			
12	CCC5PEN	0x0	RW	**CDTI5 pin enable control bit**
	CDTI5 pin enable control bit			
11	CCC4PEN	0x0	RW	**CDTI4 pin enable control bit**
	CDTI4 pin enable control bit			
10	CCC3PEN	0x0	RW	**CDTI3 pin enable control bit**
	CDTI3 pin enable control bit			
9	CCC2PEN	0x0	RW	**CDTI2 pin enable control bit**
	CDTI2 pin enable control bit			
8	CCC1PEN	0x0	RW	**CDTI1 pin enable control bit**
	CDTI1 pin enable control bit			
7	CCC0PEN	0x0	RW	**CDTI0 pin enable control bit**
	CDTI0 pin enable control bit			
6	CC6PEN	0x0	RW	**CC6 pin enable control bit**
	CC6 pin enable control bit			
5	CC5PEN	0x0	RW	**CC5 pin enable control bit**
	CC5 pin enable control bit			
4	CC4PEN	0x0	RW	**CC4 pin enable control bit**
	CC4 pin enable control bit			
3	CC3PEN	0x0	RW	**CC3 pin enable control bit**
	CC3 pin enable control bit			
2	CC2PEN	0x0	RW	**CC2 pin enable control bit**
	CC2 pin enable control bit			
1	CC1PEN	0x0	RW	**CC1 pin enable control bit**
	CC1 pin enable control bit			
0	CC0PEN	0x0	RW	**CC0 pin enable control bit**
	CC0 pin enable control bit			

25.6.161 GPIO_TIMER3_CC0ROUTE - CC0 Port/Pin Select

Offset	Bit Position																													
0x6EC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC0 pin select register	0x0	RW	**CC0 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC0 port select register	0x0	RW	**CC0 port select register**

25.6.162 GPIO_TIMER3_CC1ROUTE - CC1 Port/Pin Select

Offset	Bit Position																													
0x6F0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC1 pin select register	0x0	RW	**CC1 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC1 port select register	0x0	RW	**CC1 port select register**

25.6.163 GPIO_TIMER3_CC2ROUTE - CC2 Port/Pin Select

Offset	Bit Position																													
0x6F4	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC2 pin select register	0x0	RW	**CC2 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC2 port select register	0x0	RW	**CC2 port select register**

25.6.164 GPIO_TIMER3_CC3ROUTE - CC3 Port/Pin Select

Offset	Bit Position																													
0x6F8	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC3 pin select register	0x0	RW	**CC3 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC3 port select register	0x0	RW	**CC3 port select register**

25.6.165 GPIO_TIMER3_CC4ROUTE - CC4 Port/Pin Select

Offset	Bit Position																													
0x6FC	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC4 pin select register	0x0	RW	**CC4 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC4 port select register	0x0	RW	**CC4 port select register**

25.6.166 GPIO_TIMER3_CC5ROUTE - CC5 Port/Pin Select

Offset	Bit Position																													
0x700	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC5 pin select register	0x0	RW	**CC5 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC5 port select register	0x0	RW	**CC5 port select register**

25.6.167 GPIO_TIMER3_CC6ROUTE - CC6 Port/Pin Select

Offset	Bit Position																													
0x704	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CC6 pin select register	0x0	RW	**CC6 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CC6 port select register	0x0	RW	**CC6 port select register**

25.6.168 GPIO_TIMER3_CDTI0ROUTE - CDTI0 Port/Pin Select

Offset	Bit Position																													
0x708	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI0 pin select register	0x0	RW	**CDTI0 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI0 port select register	0x0	RW	**CDTI0 port select register**

25.6.169 GPIO_TIMER3_CDTI1ROUTE - CDTI1 Port/Pin Select

Offset	Bit Position																													
0x70C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI1 pin select register	0x0	RW	**CDTI1 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI1 port select register	0x0	RW	**CDTI1 port select register**

25.6.170 GPIO_TIMER3_CDTI2ROUTE - CDTI2 Port/Pin Select

Offset	Bit Position																													
0x710	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI2 pin select register	0x0	RW	**CDTI2 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI2 port select register	0x0	RW	**CDTI2 port select register**

25.6.171 GPIO_TIMER3_CDTI3ROUTE - CDTI3 Port/Pin Select

Offset	Bit Position																													
0x714	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI3 pin select register	0x0	RW	**CDTI3 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI3 port select register	0x0	RW	**CDTI3 port select register**

25.6.172 GPIO_TIMER3_CDTI4ROUTE - CDTI4 Port/Pin Select

Offset	Bit Position																													
0x718	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI4 pin select register	0x0	RW	**CDTI4 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI4 port select register	0x0	RW	**CDTI4 port select register**

25.6.173 GPIO_TIMER3_CDTI5ROUTE - CDTI5 Port/Pin Select

Offset	Bit Position																													
0x71C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

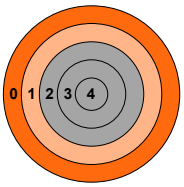
Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI5 pin select register	0x0	RW	**CDTI5 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI5 port select register	0x0	RW	**CDTI5 port select register**

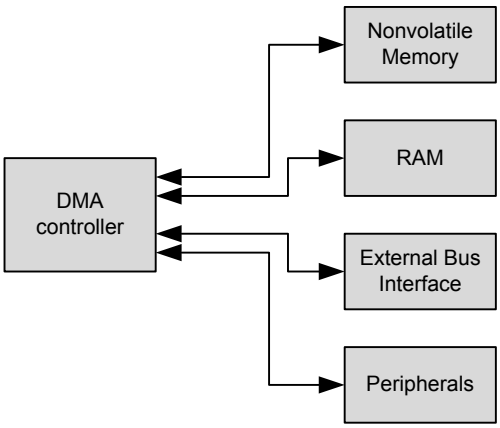
25.6.174 GPIO_TIMER3_CDTI6ROUTE - CDTI6 Port/Pin Select

Offset	Bit Position																													
0x720	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset													0x0																		0x0	
Access													RW																		RW	
Name													PIN																		PORT	

Bit	Name	Reset	Access	Description
31:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PIN CDTI6 pin select register	0x0	RW	**CDTI6 pin select register**
15:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	PORT CDTI6 port select register	0x0	RW	**CDTI6 port select register**

26. LDMA - Linked DMA Controller





Quick Facts

What?

The LDMA controller can move data without CPU intervention, effectively reducing the energy consumption for a data transfer.

Why?

The LDMA can perform data transfers more energy efficiently than the CPU and allows autonomous operation in low-energy modes.

How?

The LDMA controller has multiple highly configurable, prioritized DMA channels. A linked list of flexible descriptors makes it possible to tailor the controller to the specific needs of an application.

26.1 Introduction

The Linked Direct Memory Access (LDMA) controller performs memory transfer operations independently of the CPU. This has the benefit of reducing the energy consumption and the workload of the CPU, and enables the system to stay in low energy modes while still routing data to memory and peripherals. For example, moving data from the ADC to memory.

26.1.1 Features

- Flexible Source and Destination transfers
  - Memory-to-memory
  - Memory-to-peripheral
  - Peripheral-to-memory
  - Peripheral-to-peripheral
- DMA transfers triggered by peripherals, software, or linked list
- Single or multiple data transfers for each peripheral or software request
- Inter-channel and hardware event synchronization via trigger and wait functions
- Supports single or multiple descriptors
  - Single descriptor
  - Linked list of descriptors
  - Circular and ping-pong buffers
  - Scatter-Gather
  - Looping
  - Pause and restart triggered by other channels
  - Sophisticated flow control which can function without CPU interaction
- Channel arbitration includes:
  - Fixed priority
  - Simple round robin
  - Round robin with programmable multiple interleaved entries for higher priority requesters
- Programmable data size and source and destination address strides
- Programmable interrupt generation at the end of each DMA descriptor execution
- Little-endian/big-endian conversion
- DMA write-immediate function

26.2 Block Diagram

An overview of the LDMA and the modules it interacts with is shown in Figure 26.1. LDMA Block Diagram.

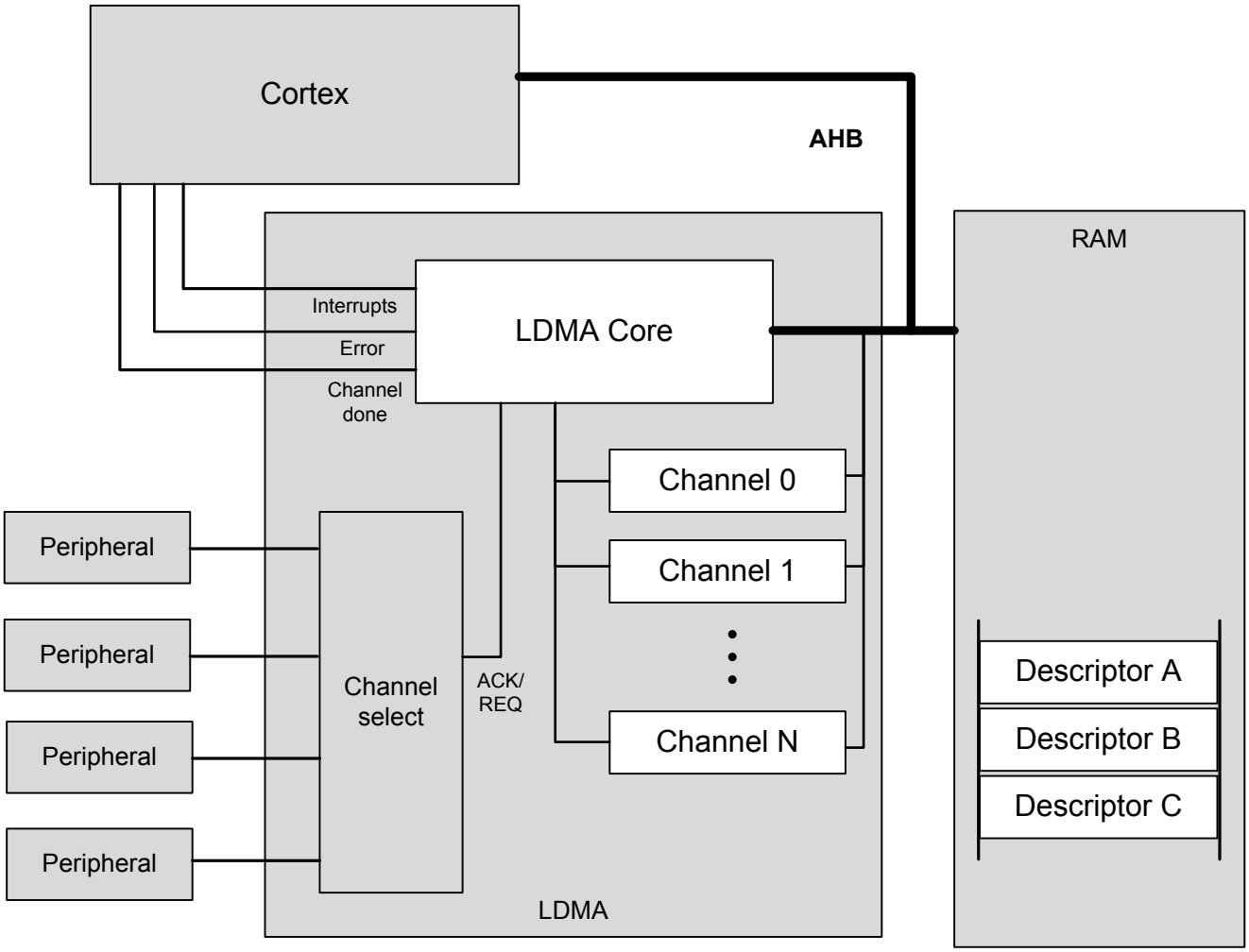


Figure 26.1. LDMA Block Diagram

The Linked DMA Controller consists of three main parts

- A DMA core that executes transfers and communicates status to the core
- A channel select block that routes peripheral DMA requests and acknowledge signals to the DMA
- A set of internal channel configuration registers for tracking the progress of each DMA channel

The DMA has access to all system memory through the AHB bus and the AHB-to-APB bridge. It can load channel descriptors from memory with no CPU intervention.

26.3 Functional Description

The Linked DMA Controller is highly flexible. It is capable of transferring data between peripherals and memory without involvement from the processor core. This can be used to increase system performance by off-loading the processor from copying large amounts of data or avoiding frequent interrupts to service peripherals needing more data or having available data. It can also be used to reduce the system energy consumption by making the LDMA work autonomously with some low energy peripherals for data transfer without having to wake up the processor core from sleep.

The Linked DMA Controller has 8 independent channels. Each of these channels can be connected to any of the available peripheral DMA transfer request input sources by writing to the channel configuration registers, see 26.3.2 Channel Configuration. In addition, each channel can also be triggered directly by software, which is useful for memory-to-memory transfers.

The channel descriptors determine what the Linked DMA Controller will do when it receives DMA transfer request. The initial descriptor is written directly to the LDMA's channel registers. If desired, the initial descriptor can link to additional linked descriptors stored in memory (RAM or Flash). Alternatively, software may also load the initial descriptor by writing the descriptor address to the LDMA_CHx_LINK register and then setting the corresponding bit the LDMA_LINKLOAD register.

Before enabling a channel, the software must take care to properly configure the channel registers including the link address and any linked descriptors. When a channel is triggered, the Linked DMA Controller will perform the memory transfers as specified by the descriptors. A descriptor contains the memory address to read from, the memory address to write to, link address of the next descriptor, the number of bytes to be transferred, etc. The channel descriptor is described in detail in 26.3.7 Channel Descriptor Data Structure.

The Linked DMA Controller supports both fixed priority and round robin arbitration. The number of fixed and round robin channels is programmable. For round robin channels, the number of arbitration slots requested for each channel is programmable. Using this scheme, it is possible to ensure that timing-critical transfers are serviced on time.

DMA transfers take place by reading a block of data at a time from the source, storing it in the LDMA's local FIFO, then writing the block out to the destination from the FIFO. Interrupts may optionally be signaled to the CPU's interrupt controller at the end of any DMA transfer or at the completion of a descriptor if the DONEIFSEN bit is set. An AHB error will always generate an interrupt.

26.3.1 Channel Descriptor

Each DMA channel has descriptor registers. A transfer can be initialized by software writing to the registers or by the DMA itself copying a descriptor from RAM to memory. When using a linked list of descriptors the first descriptor should be initialized by the CPU. The DMA itself will then copy linked descriptors to its descriptor registers as required. In addition to manually initializing the first transfer, software may also cause the LDMA to load the initial descriptor by writing the descriptor address to the LDMA_CHx_LINK register and then setting the corresponding bit the LDMA_LINKLOAD register.

The contents of the descriptor registers are dynamically updated during the DMA transfer. The contents of descriptors in memory are not edited by the controller.

Some descriptor field values are only used for linked descriptors. For example, the SRCMODE and DSTMODE bits of the LDMA_CHx_CTRL registers determine if a linked descriptor is using relative or absolute addressing. Software writes to the address registers will always use absolute addressing and never set these bits. Therefore, these bits are read only.

26.3.1.1 DMA Transfer Size

A DMA transfer is the smallest unit of data that can be transfered by the LDMA. The LDMA supports byte, half-word and word sized transfers. The SIZE field in the LDMA_CHx_CTRL register specifies the data width of one DMA transfer.

26.3.1.2 Source/Destination Increments

The SRCINC and DSTINC in the LDMA_CHx_CTRL register determines the increment between DMA transfers. The increment is in units of DMA transfers and using an increment size of 1 will transfer contiguous bytes, half-words, or words depending on the value of the SIZE field. Multiple unit increments are useful for transferring or packing/unpacking alligned data. For example using an increment of 4 with a size of BYTE will transfer word aligned bytes. An increment of 2 units with a size of HALFWORD is suitable for the transfer of word aligned half-word data. The LDMA can also pack or unpack data by using a different increment size for source and destination. For example - to convert from word aligned byte data (unpacked) to contiguous byte data (packed), set the SIZE to BYTE, SRCINC to 4, and DSTINC to 1.

SIZE may also be set to NONE which will cause the LDMA to read or write the same location for every DMA transfer. This is usefull for accessing peripheral FIFO or data registers.

26.3.1.3 Block Size

The block size defines the amount of data transferred in one arbitration. It consists of one or more DMA transfers. See 26.3.6.1 Arbitration Priority for more details.

26.3.1.4 Transfer Count

The descriptor transfer count defines how many DMA transfers to perform. The number of bytes transferred by the descripter will depend on both the transfer count XFERCNT and the SIZE field settings. TOTAL_BYTES = XFERCNT * SIZE

26.3.1.5 Descriptor List

A descriptor list consists of one or more descriptors which are executed serially. This list may be a simple sequence of descriptors, a loop of descriptors, or a combination of the two.

Each descriptor in the list can be one of several types.

- Single Transfer descriptor: Transfers TOTAL_BYTES of data and then stops.
- Linked Transfer descriptor: Transfers TOTAL_BYTES of data and then loads the next linked descriptor.
- Loop Transfer descriptor: Transfers TOTAL_BYTES of data and performs loop control (see 26.3.2.2 Loop Counter).
- Sync descriptor: Handle synchronization of the list with other entities (see 26.3.7.2 SYNC Descriptor Structure).
- WRI descriptor: Writes a value to a location in memory (see 26.3.7.3 WRI Descriptor Structure).

26.3.1.6 Addresses

Before initiating a transfer, software should write the source address, destination address, and if applicable the link address to the descriptor registers. Alternatively, software may load a descriptor from memory by writing the descriptor address to the LDMA_CHx_LINK register and setting the corresponding bit in the LDMA_LINKLOAD register.

During a DMA transfer, the DMA source and destination address registers are pointers to the next transfer address. The LDMA will update the SRC and DST addresses after each transfer. If software halts a DMA transfer by clearing the enable bit, the SRC and DST addresses will indicate the next transfer address.

When a descriptor is finished the DMA will either halt or load the next (linked) descriptor depending on the value of the LINK field in the LDMA_Chx_LINK register. After loading a linked descriptor, the descriptor registers will reflect the content of the loaded descriptor. Note that the linked descriptor must be word aligned in memory. The two least significant bits of the LDMA_CHx_LINK register are used by the LINK and LINKMODE bits. The two least significant bits of the link address are always zero.

26.3.1.7 Addressing Modes

The DMA descriptors support absolute addressing or relative addressing. When using relative addressing, the offset is relative to the current contents of the respective address registers. Regardless of the descriptor addressing modes, the address registers always indicate the absolute address. For example, when loading a descriptor using relative SRC addressing, the LDMA will add the descriptor source address (offset) to the contents of the SRCADDR register (base address). After loading, the SRCADDR register will indicate the absolute address of the loaded descriptor.

The initial descriptor must use absolute addressing. The LDMA will ignore the DSTMODE, SRCMODE, and LINKMODE bits for the initial descriptor and interpret the addresses as an absolute addresses.

Relative addressing is most useful for the link address. The initial descriptor will indicate the absolute address of the linked descriptors in memory. The linked descriptors might be an array of structures. In this case the offset between descriptors is constant and is always 4 words or 16 bytes (each descriptor has 4 words). The LINK address is not incremented or decremented after each transfer. Thus, a relative offset of 0x10 may be used for all linked descriptors.

The source and destination addresses also support relative addressing. When using relative addressing with the source or destination address registers, the LDMA adds the relative offset to the current contents of the respective address register. Since the source and destination addresses are normally incremented after each transfer, the final address will point to one unit past the last transfer. Thus, an offset of zero will give the next sequential data address.

See the example 26.4.6 2D Copy for an common use of relative addressing.

26.3.1.8 Byte Swap

Enabling byte swap reverses the endianness of the incoming source data read into the LDMA's FIFO. Byte swap is only valid for transfer sizes of word and half-word. Note that linked structure reads are not byte swapped.

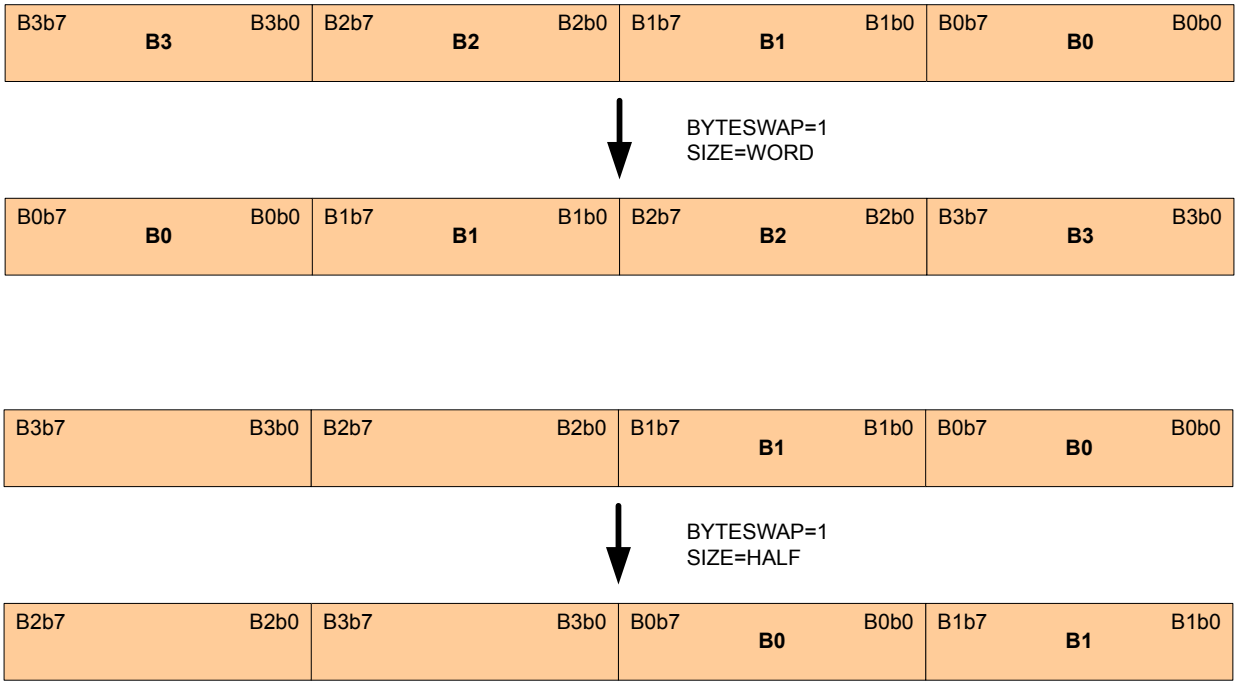


Figure 26.2. Word and Half-Word Endian Byte Swap Examples

26.3.1.9 DMA Size and Source/Destination Increment Programming

The DMA channels' SIZE, SRCINC, and DSTINC bit-fields are programmed to best utilize memory resources. They provide a means for memory packing and unpacking, as well as for matching the size of data being transmitted to or received from an IO peripheral. The following figure shows how 32-bit words of data are read from a memory source into the DMA's internal transfer FIFO, and then written out to the memory destination. The memory organization in bytes is shown as well as the first read to and write from the DMA's FIFO.

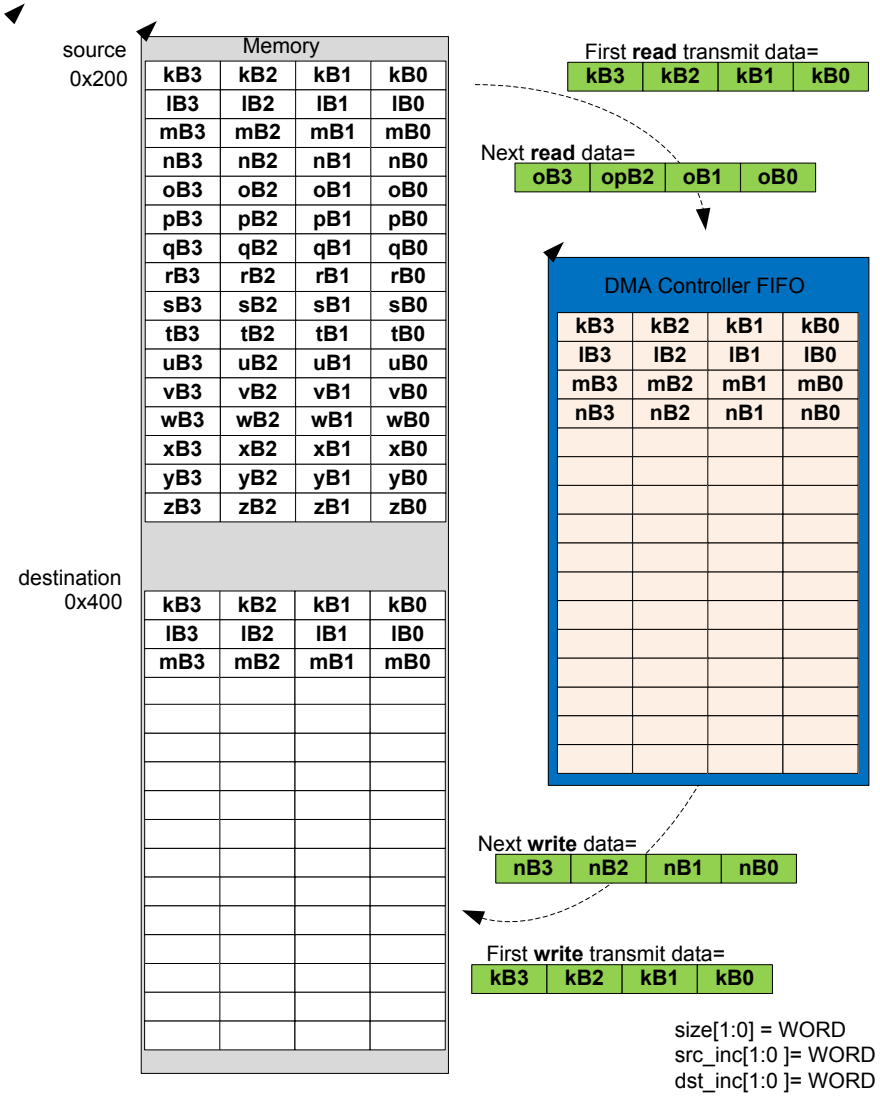


Figure 26.3. Memory-to-Memory Transfer WORD Size Example

The next example shows four variations of half-word sized transfers, with all possible combinations of half- and full-word source and destination increments. Note that when the size and source/destination increments are all configured for half-word, the resulting DMA transfer organization is equivalent to the full-word sized transfer in the previous example. The difference is that the half-word configuration requires twice as many DMA transfers.

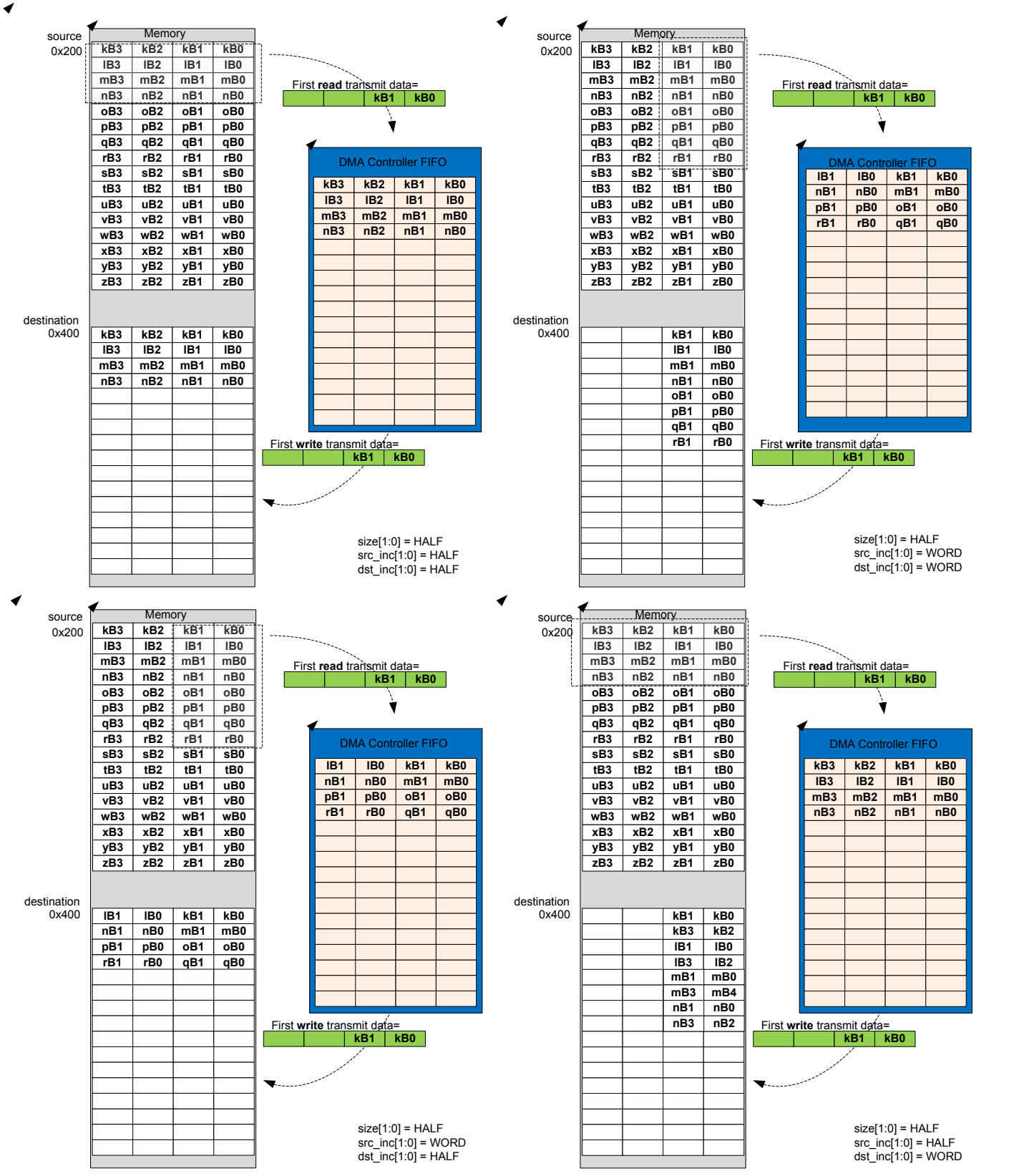


Figure 26.4. Memory-to-Memory Transfer HALF Size Examples

Fields SRCINCSIGN and DSTINCSIGN allow for address decrement. These can be used to mirror an image, for example, in the pixel copy application.

26.3.2 Channel Configuration

Each DMA channel has associated configuration and loop counter registers for controlling direction of address increment , arbitration slots, and descriptor looping.

26.3.2.1 Address Increment/Decrement

Normally DMA transfers increment the source and destination addresses after each DMA transfer. Each channel is also capable of decrementing the source and/or destination addresses after each DMA transfer. This may be useful for flipping an array or copying data from tail to head. For example, a data packet might be prepared as an array of data with increasing addresses and then transmitted from the highest address to the lowest address, from tail to head.

After reset the SRCINCSIGN and DSTINCSIGN bits in the LDMA_CHx_CFG register are cleared causing the source and destination addresses to increment after each transfer. If the SRCINCSIGN bit is set , the DMA will decrement the source address after each transfer. If the DSTINCSIGN bit in the LDMA_CHx_CFG register is set , the DMA will decrement the destination address after each transfer. Setting only one of these bits will flip the data. Setting both bits will copy from tail to head, but will not flip the data.

The SRCINCSIGN and DSTINCSIGN bits apply to all descriptors used by that channel. Software should take care to set the starting source and/or destination address to the highest data address when decrementing.

26.3.2.2 Loop Counter

Each channel has a LDMA_CHx_LOOP register that includes a loop counter field. To use looping, software should initialize the loop counter with the desired number of repetitions before enabling the transfer. A descriptor with the DECLOOPCNT bit set to TRUE will repeat the loop and decrement the loop counter until LOOPCNT = 0.

For a looping descriptor, with DECLOOPCNT=1, the LINK address in the LDMA_CHx_LINK register is used as the loop address. While LOOPCNT is greater than zero, the descriptor will execute and then the LDMA will load the next descriptor using the address specified in the LDMA_CHx_LINK register. This feature enables looping of multiple descriptors. To repeat a single descriptor, the LINK address of the descriptor should point to itself.

After LOOPCNT reaches zero, if the LINK bit in the descriptor LINK word is clear the transfer stops. If the LINK bit is set, the LDMA will load the next sequential descriptor located immediately following the looping descriptor. The behavior of the LINK bit is different for a looping descriptor. This is necessary because the LINK address is re-purposed as the loop address for a looping descriptor.

Note that LOOPCNT sets the number of repeats, not the number of iterations. The total number of loop iterations will be LOOPCNT plus 1. Normally, the LOOPCNT should be set to one or more repeats.

Also note that because there is only one LOOPCNT per channel, software intervention is required to update the LOOPCNT if a sequence of transfers contains multiple loops. It is also possible to use a write immediate DMA data transfer to update the LDMA_CHx_LOOP register.

26.3.3 Channel Select Configuration

The channel select block determines which peripheral request signal connects to each DMA channel.

This configuration is done by software through the SOURCESEL and SIGSEL fields of the LDMAXBAR_CHn_REQSEL register. SOURCESEL selects the peripheral and SIGSEL picks which DMA request signals to use from the selected peripheral. Please refer to 26.5 LDMA Source Selection Details for more information.

26.3.4 Starting a Transfer

A transfer may be started by software, a peripheral request, or a descriptor load.

Software may initiate a transfer by setting the bit for the desired channel in the LDMA_SWREQ register. In this case the channel should set SOURCESEL to NONE to prevent unintentional triggering of the channel by a peripheral.

A peripheral may trigger the channel by configuring the peripheral source and signal as described in 26.3.3 Channel Select Configuration

The LDMA may also be configured to begin a transfer immediately after a new descriptor is loaded by setting the STRUCTREQ field of the LDMA_CHx_CTRL register or descriptor word.

This configuration is done by software through the SOURCESEL and SIGSEL fields of the LDMAXBAR_CHn_REQSEL register. SOURCESEL selects the peripheral and SIGSEL picks which DMA request signals to use from the selected peripheral.

26.3.4.1 Peripheral Transfer Requests

By default peripherals issue a Single Request (SREQ) when any data is present. For peripherals with a data buffer or FIFO this occurs any time the FIFO is not empty. Upon receiving an SREQ the LDMA will perform one DMA transfer and stop till another request is made.

It is generally more efficient to wait for a peripheral to accumulate data and transfer in a burst. This both reduces overhead of the DMA engine and allows peripherals to save power by using the LDMA less often. To enable this, set the IGNORESREQ bit in the LDMA_CHx_CTRL register (or descriptor) which will cause the LDMA to ignore SREQs and wait for a full Request (REQ) signal before any data is transferred. For most peripherals with a FIFO the REQ signal is set when the FIFO is full, or a predetermined threshold has been reached. See the individual peripheral chapters for more information.

26.3.5 Managing Transfer Errors

LDMA transfer errors are normally managed using interrupts. Each LDMA channel has an ERRORx flag in the LDMA_IF register. Software should clear the ERRORx flag in the bit in the LDMA_IF register and enable error interrupts by setting the ERRORx bit in the LDMA_IEN register before initiating a DMA transfer.

The LDMA interrupt handler for a channel should check the corresponding ERROR flag bit in the LDMA_IF register. Any channel that has encountered an error will have the corresponding ERRORx flag set. If any ERROR flag bit is set, the CHERROR field in the LDMA_STATUS register will indicate the most recent channel to have generated an error. The interrupt handler should reset the channel error and clear the ERRORx flag in the LDMA_IF register before returning.

26.3.6 Arbitration

While multiple channels are configured simultaneously the LDMA engine can only be actively copying data for one channel at a time. Arbitration determines which channel is being serviced at any point in time. The LDMA will choose a channel through arbitration, transfer BLOCK_SIZE elements of that channel and then arbitrate again choosing another channel to service. This allows high priority channels to be serviced while lower priority channels are in the middle of a transfer.

26.3.6.1 Arbitration Priority

There are two modes in determining priority when the controller arbitrates: fixed priority and round robin priority.

In fixed priority mode, channel 0 has the highest priority. As the channel number increases, the priority decreases. When the LDMA controller is idle or when a transfer completes, the highest priority channel with an active request is granted the transfer. This mode guarantees smallest latency for the highest priority requesters. It is best suited for systems where peak bandwidth is well below LDMA controller's maximum ability to serve. The drawback of this mode is the possibility of starvation for lowest priority requesters.

In the round robin priority mode, each active requesting channel is serviced in the order of priority. A late arriving request on a higher priority channel will not get serviced until the next round. This mode minimizes the risk of starving low-priority latency-tolerant requesters. The drawback of this mode is higher risk of starving low-latency requesters.

The NUMFIXED field in the LDMA_CTRL register determines which channels are fixed priority and which are round robin. Channels lower than NUMFIXED are fixed priority while those above it are round robin. A value of 0x0 implies all channels are round robin. A value of 0x4 implies channels 0 through 3 are fixed priority and 4 through 7 are round robin. A value of 7 implies that channels 0 through 6 are fixed and channel 7 is round robin. This is functionally equivalent to having 8 fixed priority channels.

Fixed priority channels always take priority over round robin. As long as NUMFIXED is greater than 0, there is a possibility that a higher priority channel can starve the remaining channels.

To address the drawbacks of using fixed priority or round robin priority the LDMA implements the concept of arbitration slots. This allows for channels to have high bandwidth and low latency while preventing starvation of latency tolerant low priority channels.

Each channel has a two bit ARBSLOT field in its LDMA_CHx_CFG register. This field only applies to channels marked as round robin (determined by NUMFIXED). The channels in the same arbitration slot are treated equally with round robin scheduling. Channels marked with a higher arbitration slot will get serviced more frequently. By default all channels are placed in arbitration slot 1.

Every time the channels in slot 1 get serviced the channels in slot 2 get serviced twice, those in slot 4 get serviced 4 times, and those in slot 8 get serviced 7 times. The specific arbitration allocation can be seen by the following table. The highest arbitration slot is serviced every other arbitration cycle, allowing for low latency response. If there are no requests from channels in arbitration slot then that slot is immediately skipped.

Table 26.1. Arbitration Slot Order

Arbslot order	8	4	8	2	8	4	8	1	8	4	8	2	8	4
Arbslot1								1						
Arbslot2				1								1		
Arbslot4		1				1				1				1
Arbslot8	1		1		1		1		1		1		1	

The top row shows the order at which the arbitration slots are executed. The remaining part of the table shows a more visual interpretation of the arbitration order.

For example, if we have one low latency channel (CH0) and two latency tolerant channels (CH1 and CH2). We could use the following settings.

LDMA_CTRL.NUMFIXED = 0; set round robin for all channels.

CH0_CFG.ARBSLOTS = TWO;

CH1_CFG.ARBSLOTS = ONE;

CH2_CFG.ARBSLOTS = ONE;

If all channels are constantly requesting transfers, then the arbitration order is: CH0, CH1, CH0, CH2, CH0, CH1, CH0, CH2, CH0, etc

Note, there are no channels assigned to arbitration slot four or eight in this example, so those slots are skipped and the final sequence is ARBSLOT2, ARBSLOT1, ARBSLOT2, ARBSLOT1, etc...

Channel 1 and Channel 2 are selected in round robin order when arbitration slot 1 is executed.

If we replace the ARBSLOTS value for channel 0 with EIGHT, then the sequence would look like the following:

CH0, CH0, CH0, CH0, CH1, CH0, CH0, CH0, CH2, CH0, CH0, CH0, CH0, CH1, etc.

26.3.6.2 DMA Transfer Arbitration

In addition to the inter channel arbitration, software can configure when the controller arbitrates during a DMA transfer. This provides reduced latency to higher priority channels when configuring low priority transfers with more arbitration cycles.

The LDMA provides four bits that configure how many DMA transfers occur before it re-arbitrates. These bits are known as the BLOCKSIZE bits and they map to the arbitration rate as shown below. For example, if BLOCKSIZE = 4 then the arbitration rate is 6, that is, the controller arbitrates every 6 DMA transfers.

Table 26.2. AHB Bus Transfer Arbitration Interval lists the arbitration rates.

Table 26.2. AHB Bus Transfer Arbitration Interval

BLOCKSIZE	Arbitrate After x DMA transfers
0	x = 1
1	x = 2
2	x = 3
3	x = 4
4	x = 6
5	x = 8
6	x = 12
7	x = 16
8	x = 24
9	x = 32
10	x = 64
11	x = 128
12	x = 256
13	x = 512
14	x = 1024
15	x = lock

Note: Software must take care not to assign a low-priority channel with a large BLOCKSIZE because this prevents the controller from servicing high-priority requests, until it re-arbitrates.

The number of DMA transfers that need to be done is specified by the user in XFERCNT. When XFERCNT > BLOCKSIZE and is not an integer multiple of BLOCKSIZE then the controller always performs sequences of BLOCKSIZE transfers until XFERCNT < BLOCKSIZE remain to be transferred. The controller performs the remaining XFERCNT transfers at the end of the DMA cycle.

Software must store the value of the BLOCKSIZE bits in the channel control data structure. See 26.3.7.1 XFER Descriptor Structure for more information about the location of the BLOCKSIZE bits in the data structure.

26.3.7 Channel Descriptor Data Structure

Each channel descriptor consists of four 32-bit words:

- CTRL - control word contains information like transfer count and block size.
- SRC - source address points to where to copy data from
- DST - destination address points to where to copy data to
- LINK - link address points to where to load the next linked descriptor

These words map directly to the LDMA_CHx_CTRL, LDMA_CHx_SRC, LDMA_CHx_DST, and LDMA_CHx_LINK registers. The usage of the SRC and DST fields may differ depending on the structure type

There are three different types of descriptor data structures: **XFER, SYNC, and WRI**

26.3.7.1 XFER Descriptor Structure

This descriptor defines a typical data transfer which may be a Normal, Link, or Loop transfer.

Only this structure type can be written directly into LDMA's registers by the CPU. All descriptors may be linked to. Please refer to the register descriptions for additional information.

For specifying XFER structure type, set STRUCTTYPE to 0. Please see the peripheral register descriptions for information on the fields in this structure.

Name	Bit Position																													
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRL	DSTMODE	SRCMODE	DSTINC		SIZE		SRCINC		IGNORESREQ	DECLOOPCNT	REQMODE	DONEIFSEN	BLOCKSIZE				BYTESWAP	XFERCNT											STRUCTREQ		STRUCTTYPE	
SRC	SRCADDR																															
DST	DSTADDR																															
LINK	LINKADDR																														LINK	LINKMODE

26.3.7.2 SYNC Descriptor Structure

This descriptor defines an intra-channel synchronizing structure. It allows the channel to wait for some external stimulus before continuing on to the next descriptor. This structure is also used to provide stimulus to another channel to indicate that it may continue.

For example channel 1 may be configured to transfer a header into a buffer while channel 2 is simultaneously transferring data into the same structure. When channel 1 has completed it can wait for a sync signal from channel 2 before transferring the now complete buffer to a peripheral.

Synch descriptors do nothing until a condition is met. The condition is formed by the SYNCTRIG field in the LDMA_SYNCSTATUS register and the MATCHEN and MATCHVAL fields of the descriptor. When (SYNCTRIG & MATCHEN) == (MATCHVAL & MATCHEN) the next descriptor is loaded. In addition to waiting for the condition a Link descriptor can set or clear bits in SYNCTRIG to meet the conditions of another channel and cause it to continue. The CPU also has the ability to set and clear the SYNCTRIG bits from software.

This structure type can only be linked in from memory.

For specifying SYNC structure type, set STRUCTTYPE to 1.

Name	Bit Position																													
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRL												DONEIFSEN																			STRUCTTYPE	
SRC																	SYNCCLR								SYNCSET						
DST																	MATCHVAL								MATCHEN						
LINK	LINKADDR																													LINK	LINKMODE	

Bit	Name	Description
1:0	STRUCTTYPE	**Descriptor Type**
	This field indicates which type of descriptor this is. It must be 1 for a SYNC descriptor.	
20	DONEIFSEN	**Done if Set indicator**
	If set the interrupt flag will be set when descriptor completes.	
15:8	SYNCCLR	**Sync Trigger Clear**
	This bit-field is used to clear corresponding bits within the SYNCTRIG field of the LDMA_SYNCSTATUS register. To clear a given bit, a one should be loaded to the corresponding bit. Set is given priority over clear if both corresponding bits are loaded with a one. The sync trigger clear function can only be used when loaded from a linked structure. Alternately, the user can directly write the SYNCTRIG bit-field if required.	
7:0	SYNCSET	**Sync Trigger Set**
	This bit-field is used to set corresponding bits within the SYNCTRIG bit-field. To set a given bit, a one should be loaded to the corresponding bit. Set is given priority over clear if both corresponding bits are loaded with a one. The sync trigger set function can only be used when loaded from a linked structure. Alternately, the user can directly write the SYNCTRIG bit-field if required.	
15:8	MATCHVAL	**Sync Trigger Match Value**
	This bit-field serves as the SYNCTRIG match value. A sync match triggers the load of the next linked DMA structure as specified by link_mode, when: (SYNCTRIG & MATCHEN) == (MATCHVAL & MATCHEN).	
7:0	MATCHEN	**Sync Trigger Match Enable**
	This bit-field serves as the SYNCTRIG match enable. A sync match triggers the load of the next linked DMA structure as specified by link_mode, when: (SYNCTRIG & MATCHEN) == (MATCHVAL & MATCHEN).	

26.3.7.3 WRI Descriptor Structure

This descriptor defines a write-immediate structure. This allows a list of descriptors to write a value to a register or memory location. For example, if a channel wishes to perform two loops in a descriptor sequence a WRI may be used to program the loop count for the second loop.

This structure type can only be linked in from memory.

For specifying WRI structure type, set STRUCTTYPE to 2.

Name	Bit Position																													
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRL												DONEIFSEN																			STRUCTTYPE	
SRC	IMMVAL																															
DST	DSTADDR																															
LINK	LINKADDR																														LINK	LINKMODE

Bit	Name	Description
1:0	STRUCTTYPE	**Descriptor Type**
		This field indicates which type of descriptor this is. It must be 2 for a WRI descriptor.
20	DONEIFSEN	**Done if Set indicator**
		If set the interrupt flag will be set when descriptor completes.
31:0	IMMVAL	**Immediate Value for Write**
		This bit-field specifies the immediate data value that is to be written to the address pointed to by DSTADDR. Only one write occurs for WRI structures.
31:0	DSTADDR	**Address to write**
		This bit-field specifies the address the immediate data should be written to.

26.3.8 Interaction with the EMU

The LDMA interacts with the Energy Management Unit (EMU) to allow transfers from low energy peripherals in sleep modes. The EMU can wake up the LDMA as needed to allow data transfers to occur.

26.3.9 Interrupts

The LDMA_IF Interrupt flag register contains one DONE bit for each channel and one ERROR bit for each channel. When enabled, these interrupts are available as interrupts to the host processor. The DONE and ERROR interrupts for a channel are combined into a channel-specifc interrupt vector, LDMAx_CHNLn. If the interrupt for the DMA channel is enabled in the host processor, an interrupt will be generated if either the DONE or ERROR flag for that channel in LDMA_IF and their corresponding bits in LDMA_IEN are set.

When a descriptor finishes execution the interrupt flag for that channel will be set if the DONEIFSEN field of the LDMA_CHx_LOOP register is set. If LINK and DONEIFSEN are both set when the descriptor completes the interrupt and the linked descriptor will be immediately loaded. When the final descriptor in a linked list (LINK = 0) is finished the interrupt flag is always set regardless of the state of DONEIFSEN.

26.3.10 Debugging

For a peripheral request DMA transfer, if software sets a bit for a channel in the LDMA_DBGHALT register then the DMA will halt during a debug halt and the SRC and DST registers in the debug window will show the transfer in progress. Otherwise, during debug halt the DMA will continue to run and complete the entire transfer causing the descriptor registers to indicate the transfer has completed.

26.3.11 Multi-port Transfers

The LDMA has two bus main ports and supports transfers between them in either direction.

Which busport the LDMA uses as the source and destination for the transfer is fully configurable using the SRCBUSPORT and DSTBUSPORT bits in the CHn_CFG register.

Where to read in the next descriptor struct is configured with the STRUCTBUSPORT bit in the same register.

Short transfers between bus main ports will incur some performance loss due to synchronization and handshaking between the busports around starting and ending the transfer.

26.3.12 Extended Struct

By setting the EXTEND bit in the CHn_CTRL register the MMLDMA expects a struct size of 8 words instead of the regular 4 words.

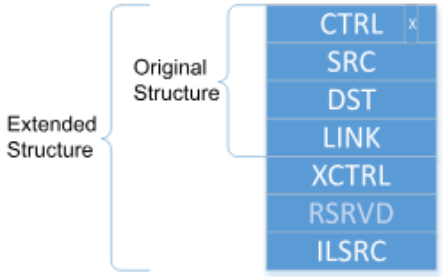


Figure 26.5. Struct and extended struct

In these additonal words, interleaving is available and the bufferable property can be configured.

26.3.13 Destination Interleaving

The LDMA supports rule-based destination interleaving, a flexible method for interleaving vectors of data which allows reading a continous stream of source data and writing it to scattered destination locations. The rules for where data is scattered or interleaved is determined by a separate table in memory with one address rule entry per unit data. This table will be read starting from the CHn_ILSRC address on the bus port that is not used for reading source data.

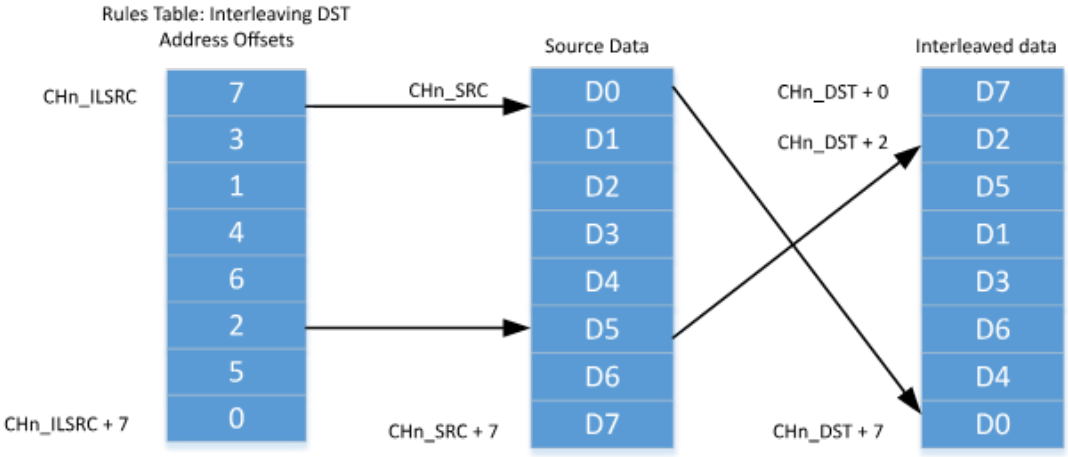


Figure 26.6. Destination Interleaving RELATIVE8 Example

Interleaving is enabled with the DSTILEN bit in CHn_XCTRL and can be applied in 3 different modes configured with the ILMODE bit field in the same register.

ABSOLUTE: An array of XFERCNT+1 32bit addresses are read from CHn_ILSRC.

```
uint32_t interleaving_address_array[xfercnt+1];
          ...
          LDMA->CH[x].XCTRL = LDMA_CH_XCTRL_DSTILEN |
          LDMA_CH_XCTRL_ILMODE_ABSOLUTE;
          LDMA->CH[x].ILSRC = interleaving_address_array;
```

RELATIVE16 An array of XFERCNT+1 16bit offsets are read from CHn_ILSRC. The offsets are applied to the DST address.

```
uint16_t interleaving_offset_array[xfercnt+1];
          ...
          LDMA->CH[x].XCTRL = LDMA_CH_XCTRL_DSTILEN |
          LDMA_CH_XCTRL_ILMODE_ABSOLUTE;
          LDMA->CH[x].ILSRC = interleaving_offset_array;
          LDMA->CH[x].DST  = start_address;
```

RELATIVE8

An array of XFERCNT+1 single byte offsets are read from CHn_ILSRC. The offsets are applied to the DST address.

```
uint8_t interleaving_offset_array[xfercnt+1];
          ...
          LDMA->CH[x].XCTRL = LDMA_CH_XCTRL_DSTILEN |
          LDMA_CH_XCTRL_ILMODE_ABSOLUTE;
          LDMA->CH[x].ILSRC = interleaving_offset_array;
          LDMA->CH[x].DST  = start_address;
```

The Interleaving configuration registers are placed in the extended part of the descriptor and therefore requires the EXTEND bit in CHn_CTRL to be set when configured through linking.

26.3.14 Disablement

Disablement is triggered by setting the EN bit in the EN register to 0. The disablement is completed when the DISABLING bit in the EN register goes low.

```
LDMA->EN_CLR = LDMA_EN_EN;           // Trigger disablement
          while (LDMA->EN & LDMA_EN_DISABLING);     // Wait for disablement to complete
```

In additon to stopping the LDMA and putting it in a low power state disablement of the LDMA will disable all channels and clear pending link load operations. All other control and configuration registers will keep their value.

Once the LDMA is disabled, registers that can be modified by the LDMA FSM will not be writable. This means that only the following registers can be written when EN=0:

- EN
- SWRST
- CTRL
- SYNCHWEN
- SYNCHWSEL
- DBGHALT
- CHn_CFG

When triggering disablement in the middle of a transfer the dma will not complete the transfer and could result in data loss and incorrect status. To ensure all transfers are fully completed disabling all channels and waiting for the ANYBUSY bit is recommended before disablement.

```
LDMA->CHDIS = _LDMA_CHDIS_MASK;        // Stop all channels
          while (LDMA->STATUS & LDMA_STATUS_ANYBUSY); // Wait until stopped

          LDMA->EN_CLR = LDMA_EN_EN;           // Trigger disablement
          while (LDMA->EN & LDMA_EN_DISABLING);     // Wait for disablement to complete
```

26.3.15 Software Reset

Software reset is triggered by writing the SWRST bit in the SWRST register. It is completed when the RESETTING bit in the SWRST register goes low.

```
LDMA->SWRST = LDMA_SWRST_SWRST;        // Trigger software reset
        while (LDMA->SWRST & LDMA_SWRST_RESETTING); // Wait for software reset to complete
```

SWRST will reset all registers in the LDMA and put it in the same state as after a power on reset.

When triggering SWRST in the middle of a transfer the dma will not complete the transfer, so disabling all channels and waiting for the ANYBUSY bit before swrst is recommended.

```
LDMA->CHDIS = _LDMA_CHDIS_MASK;        // Stop all channels
          while (LDMA->STATUS & LDMA_STATUS_ANYBUSY); // Wait until stopped

          LDMA->SWRST = LDMA_SWRST_SWRST;        // Trigger software reset
          while (LDMA->SWRST & LDMA_SWRST_RESETTING); // Wait for software reset to complete
```

26.4 Examples

This section provides examples of common LDMA usage. All examples assume the LDMA is in the reset state with the channel being configured disabled and LDMA_CHx_CFG, LDMA_CHx_LOOP, and LDMA_CHx_LINK cleared.

26.4.1 Single Direct Register DMA Transfer

This simple example uses only the Channel Descriptor registers directly and does not use linking. Software writes directly to the LDMA channel registers. This example does not use a memory based descriptor list.

This example is suitable for most simple transfers that are limited to transferring one block of data. It supports anything that can be done using a single descriptor. This includes endian conversion and packing/unpacking data. Channel 0 is used for this example.

The LDMA will be used to copy 127 contiguous half words (254 bytes) from 0x0 to 0x1000. It will allow arbitration every 4 transfers and is triggered by a CPU write to the LDMA_SWREQ register. The CH0 interrupt flag will be set when the transfer completes since the descriptor does not link to another descriptor.

- Configure LDMA_CH0_CTRL
  - DSTMODE = 0 (absolute)
  - SRCMODE = 0 (absolute)
  - SIZE = HALFWORD (16 bits)
  - DSTINC = 0 (1 half-word)
  - SRCINC = 0 (1 half-word)
  - DECLOOPCNT=0 (unused)
  - REQMODE = 1 (one request transfers all data)
  - BLOCKSIZE = 3 (4 transfers)
  - BYTESWAP=0 (no byte swap)
  - XFERCNT=127 (transfer 127 half words)
  - STRUCTTPYE=0 (TRANSFER)
- Write source address to LDMA_CH0_SRC register
- Write destination address to LDMA_CH0_DST register
- Configure the LDMAXBAR_CH0_REQSEL register for the desired peripheral or select none for a memory-to-memory transfer
- Clear and enable interrupts.
  - Write a 1 to bit 0 of the LDMA_IF_CLR register to clear the CH0 DONE flag
  - Write a 1 to bit 0 of the LDMA_IEN register to enable the CH0 interrupt
- Write a 1 to bit 0 of the LDMA_CHEN register to enable CH0

The REQMODE field is normally cleared to zero for a peripheral request transfer and will transfer the specified block size for each peripheral request. The REQMODE may be set to 1 for a memory-to-memory transfer or any time it is desired for a single DMA request to initiate complete transfer.

26.4.2 Descriptor Linked List

This example shows how to use a Linked List of descriptors. Each descriptor has a link address which points to the next descriptor in the list. A descriptor may be removed from the Linked list by altering the Link address of the one before it to point to the one after it. Descriptor Linked lists are useful when handling an array of buffers for communication data. For example, a bad packet can be removed from a receiver queue by simply removing the descriptor from the linked list.

Software loads the first descriptor into the DMA by writing the descriptor address to LDMA_CHx_LINK and setting the bit for that channel in the LDMA_LINKLOAD register. This method is preferred when using a linked list in memory since it treats the first descriptor just like all the others. However, it is also allowed for software to write the first descriptor directly to the LDMA registers.

In this example 4 descriptors are executed in series. the interrupt flag is set after the 2nd and 4th (last) descriptors have completed.

- Prepare a list of descriptors using the XFER structure type in RAM
- Initialize the CTRL, SRC, and DST members as desired
  - Setting STRUCTREQ in the CTRL word for descriptors 2-4 will cause them to begin transferring data as soon as they are loaded.
- Write 0x00000013 to the LINK member of all but the last descriptor
  - LINKMODE = 1 (relative addressing)
  - LINK = 1 (Link to the next descriptor)
  - LINKADDR = 0x00000010 (size of descriptor)
- Set the DONEIFSEN bit in the CTRL member of the 2nd structure so that the interrupt flag will be set when it completes
- Write 0x00000000 to the LINK member of the last descriptor
  - LINK = 0 (Do not link to the next descriptor)
  - LINKMODE = 0 (don't care)
  - LINKADDR = 0x00000000 (don't care)

Each descriptor now points to the start of the next descriptor as shown on the left in Figure 26.7. Descriptor Linked List. To remove a descriptor from the linked list modify the LINK address of the descriptor of the one before to point to the one after. For example to remove the third descriptor, add 0x00000010 to the LINK register of the second descriptor. The second descriptor will now point to the forth descriptor and skip over the third descriptor as shown on the right in Figure 26.7. Descriptor Linked List.

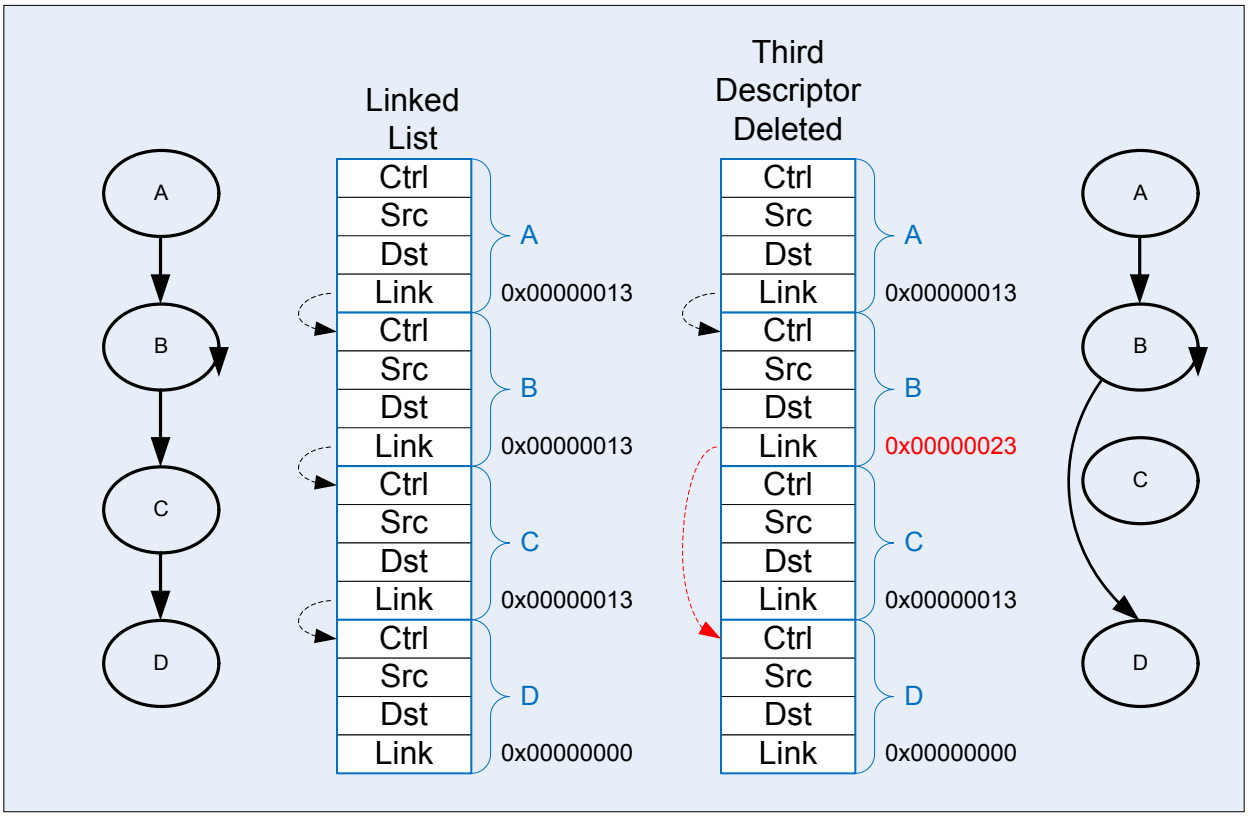


Figure 26.7. Descriptor Linked List

To start execution of the linked list of descriptors:

- Write the absolute address of the first descriptor to the LINKADR field of the LDMA_CH0_LINK register
- Set the LINK bit of LDMA_CH0_LINK register.
- Configure the LDMAXBAR_CH0_REQSEL register for the desired peripheral or select none for memory-to-memory
- Clear and enable interrupts as desired
- Set bit 0 in the LDMA_LINKLOAD register to initiate loading and execution of the first descriptor

Alternatively, software can manually copy the first descriptor contents to the LDMA_CH0_CTRL, LDMA_CH0_SRC, LDMA_CH0_DST, and LDMA_CH0_LINK registers and then enable the channel in the LDMA_CHEN register.

26.4.3 Single Descriptor Looped Transfer

This example demonstrates how to use looping using a single descriptor. This method allows a single DMA transfer to be repeated a specified number of times. The looping descriptor is stored in memory and reloaded by hardware. After a specified number of iterations, the transfer stops.

CH0 is setup to copy 4 words from the ADC FIFO into a 15 word buffer at 0x1000. It repeats 4 times to fill the entire 16 word buffer. An interrupt will fire when the entire 16 words have been transferred.

Initialize the Linked descriptor in memory as follows:

- Configure CTRL member
  - DSTMODE = 0 (absolute)
  - SRCMODE = 0 (absolute)
  - SIZE = WORD
  - DSTINC = 0 (1 WORD)
  - SRCINC = 3 (0 WORDS)
  - DECLOOPCNT=1 (decrement loop count)
  - REQMODE=1 (Use XFERCNT)
  - BLOCKSIZE = 4 (4 words)
  - BYTESWAP=0 (no swap)
  - XFERCNT= 4 (4 words)
  - STRUCTTPYE=0 (TRANSFER)
  - IGNORESREQ=1 (ignore single requests)
- Write the address ADC0_SCANDATA register to the SRC member
- Write 0x1000 address to DST member
- Configure the LINK member
  - LINK = 0 (stop after loop)
  - MODE = 1 (relative link address)
  - LINKADDR = 0 (point to ourself)
- Configure the Channel
  - Write the desired number of repeats to the LDMA_CH0_LOOP register
  - SOURCESEL in LDMAXBAR_CH0_REQSEL = ADC0 (select the ADC)
  - SIGSEL in LDMAXBAR_CH0_REQSEL = ADC0SCAN (select the scan conversion request)
- Clear and enable interrupts
- Load the descriptor using LINKLOAD as described in 26.4.2 Descriptor Linked List

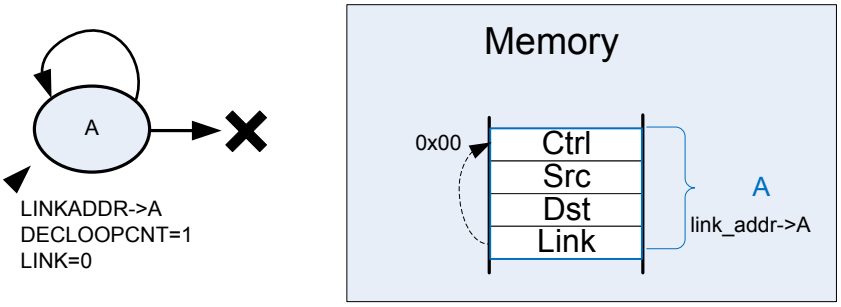


Figure 26.8. Single Descriptor Looped Transfer

Note that the looping descriptor must be stored in memory, because it must load itself from the link address in memory on each iteration.

26.4.4 Descriptor List with Looping

This example uses a descriptor list in memory with looping over multiple descriptors. This example also uses the looping feature and continues on with the next sequential descriptor after looping completes.

The descriptor list in memory is shown in figure Figure 26.9. Descriptor List with Looping. Descriptor A links to descriptor B. Descriptor B has the DECLOOPCNT bit enabled and loops back to the start of descriptor A. The LINK address of descriptor B is used for the loop address. The LINK bit is set to indicate that execution will continue after completion of looping. Once the LOOPCNT reaches zero, the LDMA will load descriptor C. Descriptor C must be located immediately following descriptor B.

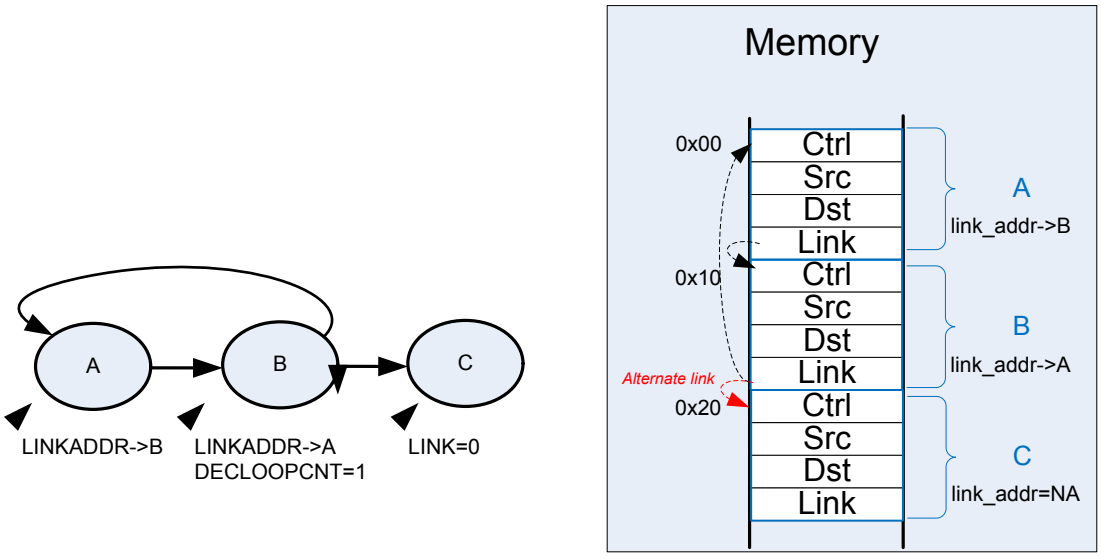


Figure 26.9. Descriptor List with Looping

Initialization is similar to the single looping descriptor with the following modifications.

- Set the LINK bit in descriptors A and B
- write the address of descriptor A into the LIKADDRESS of descriptor B
- write the address of descriptor B into the LIKADDRESS of descriptor A
- Descriptor C must be located immediately after descriptor B in memory

26.4.5 Simple Inter-Channel Synchronization

The LDMA controller features synchronization structures which allow differing channels and/or hardware events to pause a DMA sequence, and wait for a synchronizing event to restart it.

In this example DMA channel 0 and 1 are tasked with the transfer of different sets of data. Channel 0 has two transfer structures, and channel 1 just one, but channel 0 must wait until channel 1 has completed its transfer before it starts its second transfer structure.

Pausing channel 0 is accomplished by inserting a sync wait structure between the two transfer structures. This sync structure waits on SYNCTRIG[7] to be set by a sync set/clear structure which is controlled by channel 1. Sync structures do not transfer data, they can only set, clear, or wait to match the SYNCTRIG[7:0] bits. Note that sync structures cannot decrement loop counter.

```
LDMA_SYNCSTATUS
  SYNCTRIG=0x0 (at time 0)

LDMA_CH0

  Structure A @ 0x00        Structure B @ 0x10          Structure C @ 0x20
  CTRL                  CTRL                    CTRL
    STRUCTTYPE=XFER           STRUCTTYPE=SYNC           STRUCTTYPE=XFER
  LINK                  LINK                    LINK
    LINKADDR[29:0]=0x00000004      LINKADDR[29:0]=0x00000008      LINKADDR[29:0]=NA
    LINK=1                  LINK=1                  LINK=0
                      DST
                        MATCHEN=0x80
                        MATCHVAL=0x80 (waits for SYNCTRIG[7]=1)

LDMA_CH1

  Structure Y @ 0x30        Structure Z @ 0x40

  CTRL                  CTRL
    STRUCTTYPE=XFER           STRUCTTYPE=SYNC
  LINK                  LINK
    LINKADDR[29:0]=0x00000010      LINKADDR=NA
    LINK=1                  LINK=0
                      SRC
                        SRCCLR=0x0
                        SRCSET=0x80 (sets SYNCTRIG[7])
```

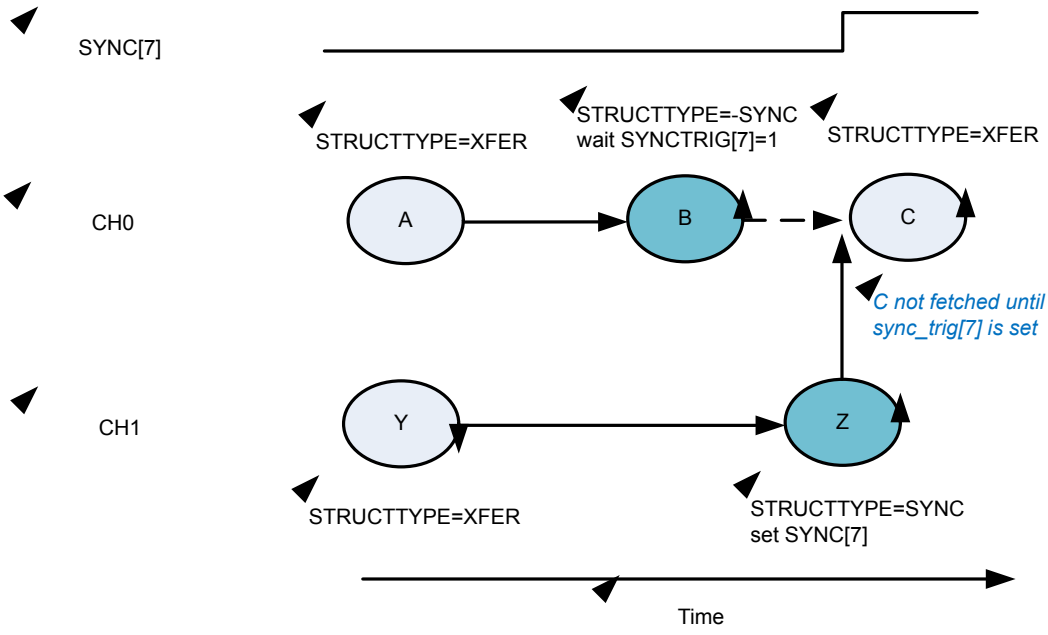


Figure 26.10. Simple Intra-channel Synchronization Example

Both A and Y effectively start at the same time. A finishes earlier, then it links to B, which waits for the SYNCTRIG[7] bit to be set before loading C. Y finishes after B is loaded, and it links to sync structure Z, which sets the SYNCTRIG[7] bit. Channel 0 responds to the trigger set by loading C for the final data transfer.

26.4.6 2D Copy

The LDMA can easily perform a 2D copy using a descriptor list with looping. This set up is visualized in Figure 26.11. 2D Copy.

For an application working with graphics, this would mean the ability to copy a rectangle of a given width and height from one picture to another.

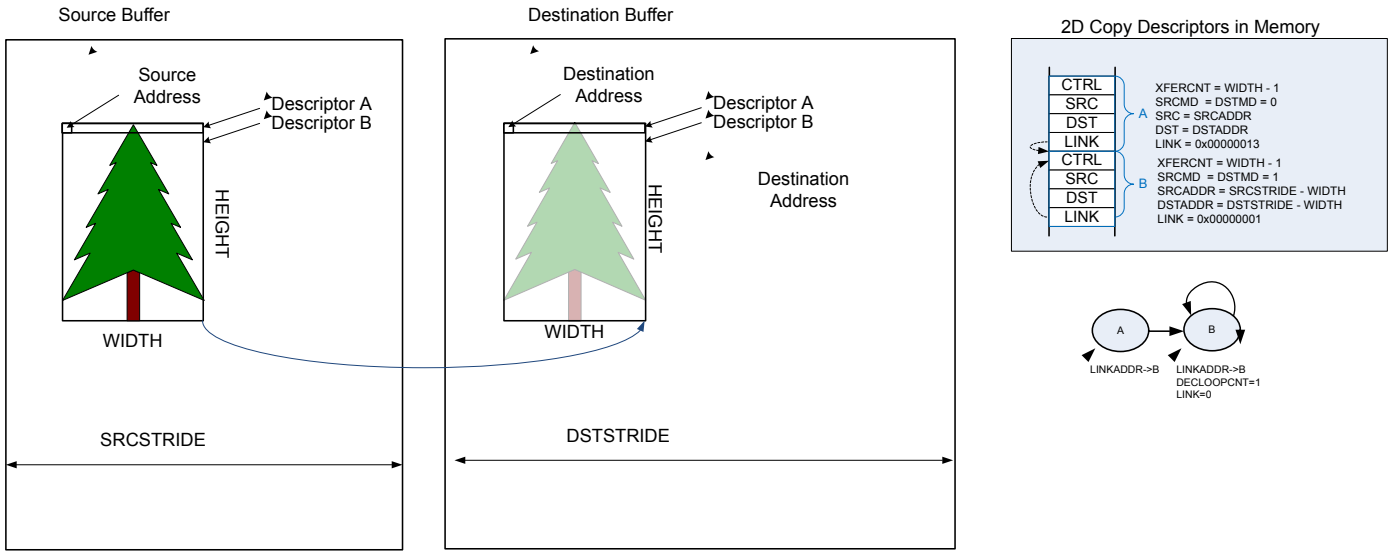


Figure 26.11. 2D Copy

The first descriptor will use absolute addressing mode and the source and destination addresses should point to the desired target addresses. The first descriptor will copy only the first row. The XFERCNT of the first descriptor is set to the desired width minus one.

- CTRL
  - XFERCNT = WIDTH - 1
  - SRCMD = 0 (absolute)
  - DSTMD = 0 (absolute)
- SRCADDR = target source address
- DSTADDR = target destination address
- LINK = 0x00000013
  - LINK=1
  - LINKMD=1
  - LINKADDR=0x00000010 (point to next descriptor)

The second descriptor will use relative addressing and the source and destination addresses are set to the desired offset. After the completion of the first descriptor, the address registers will point to the last address transferred. Thus, the width must be subtracted from the stride to get the offset. The second descriptor uses looping and the link register has not offset.

- CTRL
  - XFERCNT = WIDTH - 1
  - SRCMD = 1 (relative)
  - DSTMD = 1 (relative)
  - DECLOOPCNT = 1
- SRCADDR = desired source offset (SRCSTRIDE-WIDTH)
- DSTADDR = desired destination offset (DSTSTRIDE-WIDTH)
- LINK = 0x00000001
  - LINK=0
  - LINKMD=1 (relative)
  - LINKADDR=0x000000000 (no offset)

Because the first descriptor already transferred one row, the number of looping repeats should be the desired height minus two. Therefore, LOOPCNT should be set to HEIGHT minus two before initiating the transfer.

This same method is easily extended to copy multiple rectangles by linking descriptors together. To initialize the LDMA_CHx_LOOP register, precede each descriptor pair described above with a write immediate descriptor which writes the desired value to the LOOPCNT field of the LDMA_CHx_LOOP register.

26.4.7 Ping-Pong

Communication peripherals often use ping-pong buffers. Ping-pong buffers allow the CPU to process data in one buffer while a peripheral transmits or receives data in the other buffer.

Both transmit and receive ping-pong buffers are easily implemented using the LDMA. In either case, this requires two descriptors as shown in Figure 26.12. Infinite Ping-Pong Example. The LINKADDR field of the LINK member should point to the other descriptor. Using two adjacent descriptors and relative link addressing ensures the descriptors are easily reloadable.

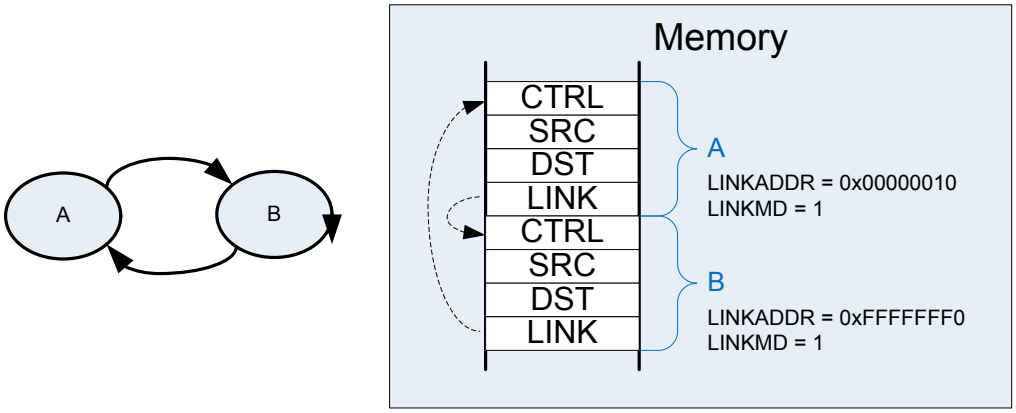


Figure 26.12. Infinite Ping-Pong Example

A **receiver** ping-pong buffer controller consists of two buffers and two descriptors stored in memory that point to the two buffers. Once initialized, as the peripheral receives data, it will fill the first buffer. Once the first buffer is full, it will link automatically to the second buffer and generate an interrupt. Software will then process the data in the first buffer while the LDMA is transferring data to the second buffer. For a receiver ping-pong buffer each descriptor should link to the other descriptor. The link bit should be set to provide infinite ping pong between the two buffers. The DONIFS bit in each descriptor should be set to generate an interrupt on the completion of each descriptor.

- Descriptor A
  - CTRL
    - DONEIFS = 1
    - other settings as desired
  - SRCADDR = peripheral source address
  - DSTADDR = memory destination address
  - LINK = 0x00000013
    - LINKADDR = 0x00000010 (next descriptor)
    - LINK = 1 (link to next descriptor)
    - LINKMD = 1 (relative addressing)
- Descriptor B
  - CTRL
    - DONEIFS = 1
    - other settings as desired
  - SRCADDR = peripheral source address
  - DSTADDR = memory destination address
  - LINK = 0xFFFFFFF3
    - LINKADDR = 0xFFFFFFF0 (previous descriptor)
    - LINK = 1 (link to previous descriptor)
    - LINKMD = 1 (relative addressing)

For **transmitter** ping-pong buffer, software will fill the first buffer and then initiate the DMA transfer. The LDMA will transmit the first buffer data while software is filling the second buffer. In this case, the two descriptors should point to each other, but not automatically continue to the second buffer. The LINK bit should be cleared to zero. Once software has loaded the first buffer, it will use the

LINKLOAD bit to load the first descriptor and transmit the data. The DONIFS need not be set in each descriptor. The DMA will stop and then generate an interrupt at the completion of each descriptor.

- Descriptor A
  - CTRL
    - DONEIFS = 0
    - other settings as desired
  - SRCADDR = memory source address
  - DSTADDR = peripheral destination address
  - LINK = 0x00000013
    - LINKADDR = 0x00000010 (next descriptor)
    - LINK = 0 (link to next descriptor)
    - LINKMD = 1 (relative addressing)
- Descriptor B
  - CTRL
    - DONEIFS = 0
    - other settings as desired
  - SRCADDR = memory source address
  - DSTADDR = peripheral destination address
  - LINK = 0xFFFFFFF3
    - LINKADDR = 0xFFFFFFF0 (previous descriptor)
    - LINK = 0 (link to previous descriptor)
    - LINKMD = 1 (relative addressing)

26.4.8 Scatter-Gather

Scatter-Gather in general refers to a process that copies data from multiple locations scattered in memory and gathers the data to a single location in memory, or vice versa. A simple descriptor list allows data gathering. For example, data from a discontiguous list of buffers might be copied to a contiguous sequential array of buffers. The inverse is also possible when a sequential array of buffers is scattered to a discontiguous list of available buffers. See section 26.4.2 Descriptor Linked List.

Some DMAs which only have two descriptors implement scatter-gather by using one descriptor to modify the other descriptor. While it is possible to implement this same behavior using the LDMA, it is much more straight-forward to just use a simple descriptor list.

26.5 LDMA Source Selection Details

26.5.1 LDMA Source Selection Details

Table 26.3. LDMA Source Selection Details

SOURCESEL	Source Name	SIGSEL	Request Signal Name
0x1	LDMAXBAR0	0x0	LDMAXBAR0PRSREQ0
		0x1	LDMAXBAR0PRSREQ1
0x2	TIMER0	0x0	TIMER0CC0
		0x1	TIMER0CC1
		0x2	TIMER0CC2
		0x3	TIMER0UFOF
0x3	TIMER1	0x0	TIMER1CC0
		0x1	TIMER1CC1
		0x2	TIMER1CC2
		0x3	TIMER1UFOF
0x4	I2C0	0x0	I2C0RXDATAV
		0x1	I2C0TXBL
0x5	I2C1	0x0	I2C1RXDATAV
		0x1	I2C1TXBL
0x6	TIMER2	0x0	TIMER2CC0
		0x1	TIMER2CC1
		0x2	TIMER2CC2
		0x3	TIMER2UFOF
		0x4	TIMER2CC3
		0x5	TIMER2CC4
		0x6	TIMER2CC5
		0x7	TIMER2CC6
0x7	TIMER3	0x0	TIMER3CC0
		0x1	TIMER3CC1
		0x2	TIMER3CC2
		0x3	TIMER3UFOF
		0x4	TIMER3CC3
		0x5	TIMER3CC4
		0x6	TIMER3CC5
		0x7	TIMER3CC6
0x8	EUSART1	0x0	EUSART1RXFL
		0x1	EUSART1TXFL
0x9	EUSART0	0x0	EUSART0RXFL
		0x1	EUSART0TXFL
0xD	ADC0	0x0	ADC0SCAN
0xE	PIXELRZ0	0x0	PIXELRZ0REQ_TXF
0xF	EUSART2	0x0	EUSART2RXFL
		0x1	EUSART2TXFL

SOURCESEL	Source Name	SIGSEL	Request Signal Name
0x10	PIXELRZ1	0x0	PIXELRZ1REQ_TXF
0x11	I2C2	0x0	I2C2RXDATAV
		0x1	I2C2TXBL

26.6 LDMA Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LDMA_IPVERSION	R	IP Version
0x004	LDMA_EN	RW ENABLE	Module Enable Disable Register
0x008	LDMA_SWRST	RW SWRST	Software Reset Register
0x00C	LDMA_CTRL	RW	Control Register
0x010	LDMA_STATUS	RH	Status Register
0x014	LDMA_SYNCSWSET	W	Sync Trig Sw Set Register
0x018	LDMA_SYNCSWCLR	W	Sync Trig Sw Clear Register
0x01C	LDMA_SYNCHWEN	RW	Sync HW Trigger Enable Register
0x020	LDMA_SYNCHWSEL	RW	Sync HW Trigger Selection Register
0x024	LDMA_SYNCSTATUS	RH	Sync Trigger Status Register
0x028	LDMA_CHEN	W	Channel Enable Register
0x02C	LDMA_CHDIS	W	Channel Disable Register
0x030	LDMA_CHSTATUS	RH	Channel Status Register
0x034	LDMA_CHBUSY	RH	Channel Busy Register
0x038	LDMA_CHDONE	RWH	Channel Linking Done Register
0x03C	LDMA_DBGHALT	RW	Channel Debug Halt Register
0x040	LDMA_SWREQ	W	Channel Software Transfer Request
0x044	LDMA_REQDIS	RW	Channel Request Disable Register
0x048	LDMA_REQPEND	RH	Channel Requests Pending Register
0x04C	LDMA_LINKLOAD	W	Channel Link Load Register
0x050	LDMA_REQCLEAR	W	Channel Request Clear Register
0x054	LDMA_IF	RWH INTFLAG	Interrupt Flag Register
0x058	LDMA_IEN	RW	Done Interrupt Enable Register
0x05C	LDMA_REQABORT	W	DMA Req Abort
0x060	LDMA_ABORTSTATUS	RH	Abort Clean Status Register
0x070	LDMA_CHx_CFG	RW	Configuration
0x074	LDMA_CHx_LOOP	RWH	Loop Counter
0x078	LDMA_CHx_CTRL	RWH	Control
0x07C	LDMA_CHx_SRC	RWH	Source Address
0x080	LDMA_CHx_DST	RWH	Destination Address
0x084	LDMA_CHx_LINK	RWH	Link Address
0x088	LDMA_CHx_XCTRL	RWH	Extended Control
0x090	LDMA_CHx_ILSRC	RWH	Interleaving Source Address
0x1000	LDMA_IPVERSION_SET	R	IP Version
0x1004	LDMA_EN_SET	RW ENABLE	Module Enable Disable Register
0x1008	LDMA_SWRST_SET	RW SWRST	Software Reset Register
0x100C	LDMA_CTRL_SET	RW	Control Register
0x1010	LDMA_STATUS_SET	RH	Status Register
0x1014	LDMA_SYNCSWSET_SET	W	Sync Trig Sw Set Register

Offset	Name	Type	Description
0x1018	LDMA_SYNCSWCLR_SET	W	Sync Trig Sw Clear Register
0x101C	LDMA_SYNCHWEN_SET	RW	Sync HW Trigger Enable Register
0x1020	LDMA_SYNCHWSEL_SET	RW	Sync HW Trigger Selection Register
0x1024	LDMA_SYNCSTATUS_SET	RH	Sync Trigger Status Register
0x1028	LDMA_CHEN_SET	W	Channel Enable Register
0x102C	LDMA_CHDIS_SET	W	Channel Disable Register
0x1030	LDMA_CHSTATUS_SET	RH	Channel Status Register
0x1034	LDMA_CHBUSY_SET	RH	Channel Busy Register
0x1038	LDMA_CHDONE_SET	RWH	Channel Linking Done Register
0x103C	LDMA_DBGHALT_SET	RW	Channel Debug Halt Register
0x1040	LDMA_SWREQ_SET	W	Channel Software Transfer Request
0x1044	LDMA_REQDIS_SET	RW	Channel Request Disable Register
0x1048	LDMA_REQPEND_SET	RH	Channel Requests Pending Register
0x104C	LDMA_LINKLOAD_SET	W	Channel Link Load Register
0x1050	LDMA_REQCLEAR_SET	W	Channel Request Clear Register
0x1054	LDMA_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1058	LDMA_IEN_SET	RW	Done Interrupt Enable Register
0x105C	LDMA_REQABORT_SET	W	DMA Req Abort
0x1060	LDMA_ABORTSTATUS_SET	RH	Abort Clean Status Register
0x1070	LDMA_CHx_CFG_SET	RW	Configuration
0x1074	LDMA_CHx_LOOP_SET	RWH	Loop Counter
0x1078	LDMA_CHx_CTRL_SET	RWH	Control
0x107C	LDMA_CHx_SRC_SET	RWH	Source Address
0x1080	LDMA_CHx_DST_SET	RWH	Destination Address
0x1084	LDMA_CHx_LINK_SET	RWH	Link Address
0x1088	LDMA_CHx_XCTRL_SET	RWH	Extended Control
0x1090	LDMA_CHx_ILSRC_SET	RWH	Interleaving Source Address
0x2000	LDMA_IPVERSION_CLR	R	IP Version
0x2004	LDMA_EN_CLR	RW ENABLE	Module Enable Disable Register
0x2008	LDMA_SWRST_CLR	RW SWRST	Software Reset Register
0x200C	LDMA_CTRL_CLR	RW	Control Register
0x2010	LDMA_STATUS_CLR	RH	Status Register
0x2014	LDMA_SYNCSWSET_CLR	W	Sync Trig Sw Set Register
0x2018	LDMA_SYNCSWCLR_CLR	W	Sync Trig Sw Clear Register
0x201C	LDMA_SYNCHWEN_CLR	RW	Sync HW Trigger Enable Register
0x2020	LDMA_SYNCHWSEL_CLR	RW	Sync HW Trigger Selection Register
0x2024	LDMA_SYNCSTATUS_CLR	RH	Sync Trigger Status Register
0x2028	LDMA_CHEN_CLR	W	Channel Enable Register
0x202C	LDMA_CHDIS_CLR	W	Channel Disable Register
0x2030	LDMA_CHSTATUS_CLR	RH	Channel Status Register
0x2034	LDMA_CHBUSY_CLR	RH	Channel Busy Register

Offset	Name	Type	Description
0x2038	LDMA_CHDONE_CLR	RWH	Channel Linking Done Register
0x203C	LDMA_DBGHALT_CLR	RW	Channel Debug Halt Register
0x2040	LDMA_SWREQ_CLR	W	Channel Software Transfer Request
0x2044	LDMA_REQDIS_CLR	RW	Channel Request Disable Register
0x2048	LDMA_REQPEND_CLR	RH	Channel Requests Pending Register
0x204C	LDMA_LINKLOAD_CLR	W	Channel Link Load Register
0x2050	LDMA_REQCLEAR_CLR	W	Channel Request Clear Register
0x2054	LDMA_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2058	LDMA_IEN_CLR	RW	Done Interrupt Enable Register
0x205C	LDMA_REQABORT_CLR	W	DMA Req Abort
0x2060	LDMA_ABORTSTATUS_CLR	RH	Abort Clean Status Register
0x2070	LDMA_CHx_CFG_CLR	RW	Configuration
0x2074	LDMA_CHx_LOOP_CLR	RWH	Loop Counter
0x2078	LDMA_CHx_CTRL_CLR	RWH	Control
0x207C	LDMA_CHx_SRC_CLR	RWH	Source Address
0x2080	LDMA_CHx_DST_CLR	RWH	Destination Address
0x2084	LDMA_CHx_LINK_CLR	RWH	Link Address
0x2088	LDMA_CHx_XCTRL_CLR	RWH	Extended Control
0x2090	LDMA_CHx_ILSRC_CLR	RWH	Interleaving Source Address
0x3000	LDMA_IPVERSION_TGL	R	IP Version
0x3004	LDMA_EN_TGL	RW ENABLE	Module Enable Disable Register
0x3008	LDMA_SWRST_TGL	RW SWRST	Software Reset Register
0x300C	LDMA_CTRL_TGL	RW	Control Register
0x3010	LDMA_STATUS_TGL	RH	Status Register
0x3014	LDMA_SYNCSWSET_TGL	W	Sync Trig Sw Set Register
0x3018	LDMA_SYNCSWCLR_TGL	W	Sync Trig Sw Clear Register
0x301C	LDMA_SYNCHWEN_TGL	RW	Sync HW Trigger Enable Register
0x3020	LDMA_SYNCHWSEL_TGL	RW	Sync HW Trigger Selection Register
0x3024	LDMA_SYNCSTATUS_TGL	RH	Sync Trigger Status Register
0x3028	LDMA_CHEN_TGL	W	Channel Enable Register
0x302C	LDMA_CHDIS_TGL	W	Channel Disable Register
0x3030	LDMA_CHSTATUS_TGL	RH	Channel Status Register
0x3034	LDMA_CHBUSY_TGL	RH	Channel Busy Register
0x3038	LDMA_CHDONE_TGL	RWH	Channel Linking Done Register
0x303C	LDMA_DBGHALT_TGL	RW	Channel Debug Halt Register
0x3040	LDMA_SWREQ_TGL	W	Channel Software Transfer Request
0x3044	LDMA_REQDIS_TGL	RW	Channel Request Disable Register
0x3048	LDMA_REQPEND_TGL	RH	Channel Requests Pending Register
0x304C	LDMA_LINKLOAD_TGL	W	Channel Link Load Register
0x3050	LDMA_REQCLEAR_TGL	W	Channel Request Clear Register
0x3054	LDMA_IF_TGL	RWH INTFLAG	Interrupt Flag Register

Offset	Name	Type	Description
0x3058	LDMA_IEN_TGL	RW	Done Interrupt Enable Register
0x305C	LDMA_REQABORT_TGL	W	DMA Req Abort
0x3060	LDMA_ABORTSTATUS_TGL	RH	Abort Clean Status Register
0x3070	LDMA_CHx_CFG_TGL	RW	Configuration
0x3074	LDMA_CHx_LOOP_TGL	RWH	Loop Counter
0x3078	LDMA_CHx_CTRL_TGL	RWH	Control
0x307C	LDMA_CHx_SRC_TGL	RWH	Source Address
0x3080	LDMA_CHx_DST_TGL	RWH	Destination Address
0x3084	LDMA_CHx_LINK_TGL	RWH	Link Address
0x3088	LDMA_CHx_XCTRL_TGL	RWH	Extended Control
0x3090	LDMA_CHx_ILSRC_TGL	RWH	Interleaving Source Address

26.7 LDMA Register Description

26.7.1 LDMA_IPVERSION - IP Version

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x3																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x3	R	**IPVERSION**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

26.7.2 LDMA_EN - Module Enable Disable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status** When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFO
0	EN	0x0	RW	**Module Enable** The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.

26.7.3 LDMA_SWRST - Software Reset Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Software Reset Busy Status** When SWRST command is issued, resetting logic sets RESETTING status immediately, and later it is cleared when reset process finishes
0	SWRST	0x0	W	**Software Reset Command** A software reset command field resets the module back to the initial condition, similar to a power on reset condition

26.7.4 LDMA_CTRL - Control Register

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset			0x1E																													
Access			RW																													
Name			NUMFIXED																													

Bit	Name	Reset	Access	Description
31:30	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
29:24	NUMFIXED	0x1E	RW	**Number of Fixed Priority Channels** This field defines the number of Fixed Priority Arbitration channels. Channels CH0 though CH(n-1) are fixed, and channels CH(n) through CH<max> are round robin, where n is the field value. The reset value will give all fixed channels.
23:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

26.7.5 LDMA_STATUS - Status Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x8																	0x0					0x7					0x0	0x0
Access				R																	R					R					R	R
Name				CHNUM																	CHERROR					CHGRANT					ANYREQ	ANYBUSY

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28:24	CHNUM	0x8	R	**Number of Channels**
	The value of CHNUM always reads the total number of channels present for this instance of the DMA controller module.			
23:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11:8	CHERROR	0x0	R	**Errant Channel Number**
	When the ERROR flag is set in the LDMA_IF register, the CHERROR field will indicate the most recent channel to have a transfer error.			
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6:3	CHGRANT	0x7	R	**Granted Channel Number**
	The value of this field indicates the currently active channel or last active channel. Note that the reset value for this field is CHNUM-1 to allow round robin arbitration to start at channel 0 after reset.			
2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	ANYREQ	0x0	R	**Any DMA Channel Request Pending**
	The value of this bit will be TRUE (1) if any requests are pending			
0	ANYBUSY	0x0	R	**Any DMA Channel Busy**
	The value of this bit will be TRUE (1) if one or more DMA channels are actively transferring data			

26.7.6 LDMA_SYNCSWSET - Sync Trig Sw Set Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W(nB)							
Name																									SYNCSWSET							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SYNCSWSET	0x0	W(nB)	**DMA SYNC Software Trigger Set**
	Sets the corresponding bit in the SYNCSTATUS.SYNCTRIG field to value 1.			

26.7.7 LDMA_SYNCSWCLR - Sync Trig Sw Clear Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W(nB)							
Name																									SYNCSWCLR							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SYNCSWCLR	0x0	W(nB)	**DMA SYNC Software Trigger Clear**
	Clears the corresponding bit in the SYNCSTATUS.SYNCTRIG field to value 0.			

26.7.8 LDMA_SYNCHWEN - Sync HW Trigger Enable Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																0x0							
Access									RW																RW							
Name									SYNCCLREN																SYNCSETEN							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:16	SYNCCLREN	0x0	RW	**Hardware Sync Trigger Clear Enable**
	Enables the corresponding bit in the SYNCSTATUS.SYNCTRIG field to be cleared by PRS channel 7-0, mapping to bits [23:16].			
15:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SYNCSETEN	0x0	RW	**Hardware Sync Trigger Set Enable**
	Enables the corresponding bit in the SYNCSTATUS.SYNCTRIG field to be set by PRS channel 7-0, mapping to bits [7:0].			

26.7.9 LDMA_SYNCHWSEL - Sync HW Trigger Selection Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0																0x0							
Access									RW																RW							
Name									SYNCCLREDGE																SYNCSETEDGE							

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23:16	SYNCCLREDGE	0x0	RW	**Hardware Sync Trigger Clear Edge Select**
	Select rising or falling edge detection on PRS to clear trigger.			
	Value	Mode		Description
	0	RISE		Use rising edge detection
	1	FALL		Use falling edge detection
15:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SYNCSETEDGE	0x0	RW	**Hardware Sync Trigger Set Edge Select**
	Select rising or falling edge detection on PRS to set trigger.			
	Value	Mode		Description
	0	RISE		Use rising edge detection
	1	FALL		Use falling edge detection

26.7.10 LDMA_SYNCSTATUS - Sync Trigger Status Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									SYNCTRIG							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SYNCTRIG	0x0	R	**sync trig status**
	Reflects the status of setting and clearing by software (SYNCSWSET/SYNCSWCLR), hardware (PRS), and loading SYNC structures. Setting a bit always takes prescedence over clearing.			

26.7.11 LDMA_CHEN - Channel Enable Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W(nB)							
Name																									CHEN							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	CHEN	0x0	W(nB)	**Channel Enables** Setting one of these bits will enable the respective DMA channel, writing zeros has no effect

26.7.12 LDMA_CHDIS - Channel Disable Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W(nB)							
Name																									CHDIS							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	CHDIS	0x0	W(nB)	**DMA Channel disable** Setting one of these bits will disable respective DMA channel, writing zeros has no effect. If set while a transfer is in progress, the current transfer block will complete. The remaining blocks will pause until resumed later by setting corresponding CHEN bit.

26.7.13 LDMA_CHSTATUS - Channel Status Register

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									CHSTATUS							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	CHSTATUS	0x0	R	**DMA Channel Status**
	Each bit represents the enabled status of the channel, channel n is enabled when bit n is 1 and disabled when bit n is 0.			

26.7.14 LDMA_CHBUSY - Channel Busy Register

Offset	Bit Position																													
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									BUSY							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	BUSY	0x0	R	**Channels Busy**
	The bits of this field read 1 when the corresponding channel is busy. This bit will be set when the channel starts a transfer and cleared when the tranfer completes.			

26.7.15 LDMA_CHDONE - Channel Linking Done Register

Offset	Bit Position																													
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access																									RW	RW	RW	RW	RW	RW	RW	RW
Name																									CHDONE7	CHDONE6	CHDONE5	CHDONE4	CHDONE3	CHDONE2	CHDONE1	CHDONE0

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	CHDONE7	0x0	RW	**Channel 7 Linking Done Status flag** Each DMA channel sets the corresponding bit in this register when the entire linked transfer is done and there is no more linking to do. The interrupt service routine should clear these bits. For non-link structures this will be set once the structure is done.
6	CHDONE6	0x0	RW	**Channel 6 Linking Done Status flag** Each DMA channel sets the corresponding bit in this register when the entire linked transfer is done and there is no more linking to do. The interrupt service routine should clear these bits. For non-link structures this will be set once the structure is done.
5	CHDONE5	0x0	RW	**Channel 5 Linking Done Status flag** Each DMA channel sets the corresponding bit in this register when the entire linked transfer is done and there is no more linking to do. The interrupt service routine should clear these bits. For non-link structures this will be set once the structure is done.
4	CHDONE4	0x0	RW	**Channel 4 Linking Done Status flag** Each DMA channel sets the corresponding bit in this register when the entire linked transfer is done and there is no more linking to do. The interrupt service routine should clear these bits. For non-link structures this will be set once the structure is done.
3	CHDONE3	0x0	RW	**Channel 3 Linking Done Status flag** Each DMA channel sets the corresponding bit in this register when the entire linked transfer is done and there is no more linking to do. The interrupt service routine should clear these bits. For non-link structures this will be set once the structure is done.
2	CHDONE2	0x0	RW	**Channel 2 Linking Done Status flag** Each DMA channel sets the corresponding bit in this register when the entire linked transfer is done and there is no more linking to do. The interrupt service routine should clear these bits. For non-link structures this will be set once the structure is done.
1	CHDONE1	0x0	RW	**Channel 1 Linking Done Status flag** Each DMA channel sets the corresponding bit in this register when the entire linked transfer is done and there is no more linking to do. The interrupt service routine should clear these bits. For non-link structures this will be set once the structure is done.
0	CHDONE0	0x0	RW	**Channel 0 Linking Done Status flag** Each DMA channel sets the corresponding bit in this register when the entire linked transfer is done and there is no more linking to do. The interrupt service routine should clear these bits. For non-link structures this will be set once the structure is done.

26.7.16 LDMA_DBGHALT - Channel Debug Halt Register

Offset	Bit Position																													
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									DBGHALT							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	DBGHALT	0x0	RW	**DMA Debug Halt**
	Setting one of these bits will mask the corresponding DMA channel's peripheral request when debugging and the CPU is halted. This may be useful for debugging DMA software.			

26.7.17 LDMA_SWREQ - Channel Software Transfer Request

Offset	Bit Position																													
0x040	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W(nB)							
Name																									SWREQ							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	SWREQ	0x0	W(nB)	**Software Transfer Requests**
	Setting one of these bits will trigger a DMA transfer for the corresponding channel. Writing zeros has no effect.			

26.7.18 LDMA_REQDIS - Channel Request Disable Register

Offset	Bit Position																													
0x044	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									REQDIS							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	REQDIS	0x0	RW	**DMA Request Disables** Setting one of these bits will disable peripheral requests for the corresponding channel. When cleared any pending peripheral requests will be serviced.

26.7.19 LDMA_REQPEND - Channel Requests Pending Register

Offset	Bit Position																													
0x048	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									R							
Name																									REQPEND							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	REQPEND	0x0	R	**DMA Requests Pending** When a DMA channel has a pending request, the corresponding REQPEND bit will read 1.

26.7.20 LDMA_LINKLOAD - Channel Link Load Register

Offset	Bit Position																													
0x04C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W(nB)							
Name																									LINKLOAD							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	LINKLOAD	0x0	W(nB)	**DMA Link Loads**
	Setting one of these bits will force the corresponding DMA channel to load the next DMA structure and enable the channel. This empowers software to step through a sequence of descriptors. When this bit is set, the current configuration of the channel is ignored.			

26.7.21 LDMA_REQCLEAR - Channel Request Clear Register

Offset	Bit Position																													
0x050	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W(nB)							
Name																									REQCLEAR							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	REQCLEAR	0x0	W(nB)	**DMA Request Clear**
	Setting one of these bits will clear any internally registered transfer requests for the corresponding channel.			

26.7.22 LDMA_IF - Interrupt Flag Register

Offset	Bit Position																													
0x054	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset									0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0									0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access									RW	RW	RW	RW	RW	RW	RW	RW									RW	RW	RW	RW	RW	RW	RW	RW
Name									ERROR7	ERROR6	ERROR5	ERROR4	ERROR3	ERROR2	ERROR1	ERROR0									DONE7	DONE6	DONE5	DONE4	DONE3	DONE2	DONE1	DONE0

Bit	Name	Reset	Access	Description
31:24	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
23	ERROR7	0x0	RW	**Error Flag**
	Set to 1 on an Error			
22	ERROR6	0x0	RW	**Error Flag**
	Set to 1 on an Error			
21	ERROR5	0x0	RW	**Error Flag**
	Set to 1 on an Error			
20	ERROR4	0x0	RW	**Error Flag**
	Set to 1 on an Error			
19	ERROR3	0x0	RW	**Error Flag**
	Set to 1 on an Error			
18	ERROR2	0x0	RW	**Error Flag**
	Set to 1 on an Error			
17	ERROR1	0x0	RW	**Error Flag**
	Set to 1 on an Error			
16	ERROR0	0x0	RW	**Error Flag**
	Set to 1 on an Error			
15:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	DONE7	0x0	RW	**Done Interrupt Flag**
	This bit is set when the corresponding channel completes a transfer or sync operation if either its CTRL.DONESET bit is 1 or it is the last descriptor in the chain (i.e. it is not linking).			
6	DONE6	0x0	RW	**Done Interrupt Flag**
	This bit is set when the corresponding channel completes a transfer or sync operation if either its CTRL.DONESET bit is 1 or it is the last descriptor in the chain (i.e. it is not linking).			
5	DONE5	0x0	RW	**Done Interrupt Flag**
	This bit is set when the corresponding channel completes a transfer or sync operation if either its CTRL.DONESET bit is 1 or it is the last descriptor in the chain (i.e. it is not linking).			
4	DONE4	0x0	RW	**Done Interrupt Flag**
	This bit is set when the corresponding channel completes a transfer or sync operation if either its CTRL.DONESET bit is 1 or it is the last descriptor in the chain (i.e. it is not linking).			
3	DONE3	0x0	RW	**Done Interrupt Flag**
	This bit is set when the corresponding channel completes a transfer or sync operation if either its CTRL.DONESET bit is 1 or it is the last descriptor in the chain (i.e. it is not linking).			

Bit	Name	Reset	Access	Description
2	DONE2	0x0	RW	**Done Interrupt Flag**
	This bit is set when the corresponding channel completes a transfer or sync operation if either its CTRL.DONESET bit is 1 or it is the last descriptor in the chain (i.e. it is not linking).			
1	DONE1	0x0	RW	**Done Interrupt Flag**
	This bit is set when the corresponding channel completes a transfer or sync operation if either its CTRL.DONESET bit is 1 or it is the last descriptor in the chain (i.e. it is not linking).			
0	DONE0	0x0	RW	**Done Interrupt Flag**
	This bit is set when the corresponding channel completes a transfer or sync operation if either its CTRL.DONESET bit is 1 or it is the last descriptor in the chain (i.e. it is not linking).			

26.7.23 LDMA_IEN - Done Interrupt Enable Register

Offset	Bit Position																													
0x058	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																								0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
Access	RW																								RW	RW	RW	RW	RW	RW	RW	RW
Name	ERROR																								DONE7	DONE6	DONE5	DONE4	DONE3	DONE2	DONE1	DONE0

Bit	Name	Reset	Access	Description
31	ERROR	0x0	RW	**Error Interrupt Enable**
	This is the bitfield to enable the AHB bus error interrupt			
30:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	DONE7	0x0	RW	**Done Interrupt Enable**
	Enables done interrupts			
6	DONE6	0x0	RW	**Done Interrupt Enable**
	Enables done interrupts			
5	DONE5	0x0	RW	**Done Interrupt Enable**
	Enables done interrupts			
4	DONE4	0x0	RW	**Done Interrupt Enable**
	Enables done interrupts			
3	DONE3	0x0	RW	**Done Interrupt Enable**
	Enables done interrupts			
2	DONE2	0x0	RW	**Done Interrupt Enable**
	Enables done interrupts			
1	DONE1	0x0	RW	**Done Interrupt Enable**
	Enables done interrupts			
0	DONE0	0x0	RW	**Done Interrupt Enable**
	Enables done interrupts			

26.7.24 LDMA_REQABORT - DMA Req Abort

Offset	Bit Position																													
0x05C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									W(nB)							
Name																									REQABORT							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	REQABORT DMA REQ ABORT	0x0	W(nB)	**REQ Abort**

26.7.25 LDMA_ABORTSTATUS - Abort Clean Status Register

Offset	Bit Position																													
0x060	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0xFF							
Access																									R							
Name																									ABORTSTATUS							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	ABORTSTATUS This bit is updated when ReqAbort is complete.	0xFF	R	**Req Abort Status bit**

26.7.26 LDMA_CHx_CFG - Configuration

Offset	Bit Position																															
0x070	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset								0x0	0x0	0x0	0x0	0x0			0x0																	
Access								RW	RW	RW	RW	RW			RW																	
Name								DSTBUSPORT	SRCBUSPORT	STRUCTBUSPORT	DSTINCSIGN	SRCINCSIGN			ARBSLOTS																	

Bit	Name	Reset	Access	Description
31:25	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
24	DSTBUSPORT	0x0	RW	**Destination Bus Port**
	Specifies the DST operation bus port			
	Value	Mode		Description
	0	AHBM0		
	1	AHBM1		
23	SRCBUSPORT	0x0	RW	**Source Bus Port**
	Specifies the SRC operation bus port			
	Value	Mode		Description
	0	AHBM0		
	1	AHBM1		
22	STRUCTBUSPORT	0x0	RW	**Structure Fetch Bus Port**
	Specifies the Structure Fetch operation bus port			
	Value	Mode		Description
	0	AHBM0		
	1	AHBM1		
21	DSTINCSIGN	0x0	RW	**Destination Address Increment Sign**
	0: Increment destination address, 1: Decrement destination address			
	Value	Mode		Description
	0	POSITIVE		Increment destination address
	1	NEGATIVE		Decrement destination address
20	SRCINCSIGN	0x0	RW	**Source Address Increment Sign**
	0: Increment source address, 1: Decrement source address			
	Value	Mode		Description
	0	POSITIVE		Increment source address
	1	NEGATIVE		Decrement source address

Bit	Name	Reset	Access	Description
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	ARBSLOTS	0x0	RW	**Arbitration Slot Number Select**
	For channels using round robin arbitration, this bit-field is used to select the number of slots in the round robin queue.			

Value	Mode	Description
0	ONE	One arbitration slot selected
1	TWO	Two arbitration slots selected
2	FOUR	Four arbitration slots selected
3	EIGHT	Eight arbitration slots selected

15:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

26.7.27 LDMA_CHx_LOOP - Loop Counter

Offset	Bit Position																													
0x074	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0							
Access																									RW							
Name																									LOOPCNT							

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	LOOPCNT	0x0	RW	**Linked Structure Sequence Loop Counter**
	This bit-field specifies the number of repetitions when using looping descriptors. Software should write to LOOPCNT before using a looping descriptor.			

26.7.28 LDMA_CHx_CTRL - Control

Offset	Bit Position																													
0x078	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0	0x0	0x0		0x0		0x0		0x0	0x0	0x0	0x0	0x0				0x0	0x0											0x0	0x0	0x0	
Access	R	R	RW		RW		RW		RW	RW	RW	RW	RW				RW	RW											R	RW	RW	
Name	DSTMODE	SRCMODE	DSTINC		SIZE		SRCINC		IGNORESREQ	DECLOOPCNT	REQMODE	DONEIEN	BLOCKSIZE				BYTESWAP	XFERCNT											STRUCTREQ	EXTEND	STRUCTTYPE	

Bit	Name	Reset	Access	Description
31	DSTMODE	0x0	R	**Destination Addressing Mode**

This field specifies the destination addressing mode of linked descriptors. After loading a linked descriptor, reading this field will indicate the destination addressing mode of the linked descriptor. Note that the first descriptor always uses absolute addressing mode.

Value	Mode	Description
0	ABSOLUTE	The DSTADDR field of LDMA_CHx_DST contains the absolute address of the destination data.
1	RELATIVE	The DSTADDR field of LDMA_CHx_DST contains the relative offset of the destination data.

Bit	Name	Reset	Access	Description
30	SRCMODE	0x0	R	**Source Addressing Mode**

This field specifies the source addressing mode of linked descriptors. After loading a linked descriptor, reading this field will indicate the source addressing mode of the linked descriptor. Note that the first descriptor always uses absolute addressing mode.

Value	Mode	Description
0	ABSOLUTE	The SRCADDR field of LDMA_CHx_SRC contains the absolute address of the source data.
1	RELATIVE	The SRCADDR field of LDMA_CHx_SRC contains the relative offset of the source data.

Bit	Name	Reset	Access	Description
29:28	DSTINC	0x0	RW	**Destination Address Increment Size**

This bit-field specifies the stride or number of unit data addresses to increment the destination address after each unit of data is transferred. The unit data width is controlled by the SIZE bit-field and can be a byte, half-word or word.

Value	Mode	Description
0	ONE	Increment destination address by one unit data size after each write
1	TWO	Increment destination address by two unit data sizes after each write
2	FOUR	Increment destination address by four unit data sizes after each write
3	NONE	Do not increment the destination address. Writes are made to a fixed destination address, for example writing to a FIFO.

Bit	Name	Reset	Access	Description
27:26	SIZE	0x0	RW	**Unit Data Transfer Size**

This field specifies the size of data transferred.

Value	Mode	Description

Bit	Name	Reset	Access	Description
	0	BYTE		Each unit transfer is a byte
	1	HALFWORD		Each unit transfer is a half-word
	2	WORD		Each unit transfer is a word
25:24	SRCINC	0x0	RW	**Source Address Increment Size**
	This bit-field specifies the stride or number of unit data addresses to increment the source address after each unit of data is transferred. The unit data width is controlled by the SIZE bit-field and can be a byte, half-word or word.			
	Value	Mode		Description
	0	ONE		Increment source address by one unit data size after each read
	1	TWO		Increment source address by two unit data sizes after each read
	2	FOUR		Increment source address by four unit data sizes after each read
	3	NONE		Do not increment the source address. In this mode reads are made from a fixed source address, for example reading FIFO.
23	IGNORESREQ	0x0	RW	**Ignore Sreq**
	The channel arbiter will ignore single requests (SREQ) and only respond to multiple requests (REQ) when this bit is set.			
22	DECLOOPCNT	0x0	RW	**Decrement Loop Count**
	When using looping, setting this bit will decrement the LOOPCNT field in the LDMA_CHx_LOOP register after each descriptor execution.			
21	REQMODE	0x0	RW	**DMA Request Transfer Mode Select**
	Selects the DMA Request Transfer mode.			
	Value	Mode		Description
	0	BLOCK		The LDMA transfers one BLOCKSIZE per transfer request.
	1	ALL		One transfer request transfers all units as defined by the XFRCNT field.
20	DONEIEN	0x0	RW	**DMA Operation Done Interrupt Flag Set**
	When enabled, the corresponding DONEIF bit will get set when descriptor processing is done (i.e. when transfer completes, next descriptor is linked or a syncmatch is hit in the case of SYNC transfers).			
19:16	BLOCKSIZE	0x0	RW	**Block Transfer Size**
	This bit-field controls the number of unit data transfers per arbitration cycle			
	Value	Mode		Description
	0	UNIT1		1 unit transfer per arbitration
	1	UNIT2		2 unit transfers per arbitration
	2	UNIT3		3 unit transfers per arbitration
	3	UNIT4		4 unit transfers per arbitration
	4	UNIT6		6 unit transfers per arbitration
	5	UNIT8		8 unit transfers per arbitration
	6	UNIT12		12 unit transfers per arbitration
	7	UNIT16		16 unit transfers per arbitration
	8	UNIT24		24 unit transfers per arbitration
	9	UNIT32		32 unit transfers per arbitration
	10	UNIT64		64 unit transfers per arbitration

Bit	Name	Reset	Access	Description
	11	UNIT128		128 unit transfers per arbitration
	12	UNIT256		256 unit transfers per arbitration
	13	UNIT512		512 unit transfers per arbitration
	14	UNIT1024		1024 unit transfers per arbitration
	15	ALL		Transfer all units as specified by the XFRCNT field
15	BYTESWAP	0x0	RW	**Endian Byte Swap**
	For word and half-word transfers, setting this bit will swap all bytes of each word or half-word.			
14:4	XFERCNT	0x0	RW	**DMA Unit Data Transfer Count**
	Specifies number of unit data (words, half-words, or bytes) to transfer, as determined by the SIZE field. The value written should be one less than the desired transfer count.			
3	STRUCTREQ	0x0	R	**Structure DMA Transfer Request**
	Structure Transfer Request			
2	EXTEND	0x0	RW	**Extend**
	Extend data structure			
1:0	STRUCTTYPE	0x0	RW	**DMA Structure Type**
	DMA Structure type			

Value	Mode	Description
0	TRANSFER	DMA transfer structure type selected.
1	SYNCHRONIZE	Synchronization structure type selected.
2	WRITE	Write immediate value structure type selected.

26.7.29 LDMA_CHx_SRC - Source Address

Offset	Bit Position																															
0x07C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	ADDR																															

Bit	Name	Reset	Access	Description
31:0	ADDR	0x0	RW	**Source Data Address**
	Writing to this register sets the source address. Reading from this register during a DMA transfer will indicate the next source read address. The value of this register is incremented or decremented with each source read.			

26.7.30 LDMA_CHx_DST - Destination Address

Offset	Bit Position																															
0x080	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	ADDR																															

Bit	Name	Reset	Access	Description
31:0	ADDR	0x0	RW	**Destination Data Address**
	Writing to this register sets the destination address. Reading from this register during a DMA transfer will indicate the next destination write address. This value of this register is incremented or decremented with each destination write.			

26.7.31 LDMA_CHx_LINK - Link Address

Offset	Bit Position																															
0x084	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																														0x0	0x0
Access	RW																														RW	R
Name	LINKADDR																														LINK	LINKMODE

Bit	Name	Reset	Access	Description
31:2	LINKADDR	0x0	RW	**Link Structure Address**
	To use linking, write the address of the the first linked descriptor to this register. When a linked descriptor is loaded, it may also be linked to another descriptor. Reading this register will reflect the address of the next linked descriptor.			
1	LINK	0x0	RW	**Link Next Structure**
	After completing the initial transfer, if this bit is set, the DMA will load the next linked descriptor. If the next linked descriptor also has this bit set, the DMA will load the next linked descriptor.			
0	LINKMODE	0x0	R	**Link Structure Addressing Mode**
	This field specifies the addressing mode of linked descriptors. After loading a linked descriptor, reading this field will indicate the addressing mode of the loaded linked descriptor. Note that the first descriptor always uses absolute addressing mode.			

Value	Mode	Description
0	ABSOLUTE	The LINKADDR field of LDMA_CHx_LINK contains the absolute address of the linked descriptor.
1	RELATIVE	The LINKADDR field of LDMA_CHx_LINK contains the relative offset of the linked descriptor.

26.7.32 LDMA_CHx_XCTRL - Extended Control

Offset	Bit Position																															
0x088	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0	0x0		0x0				
Access																									RW	RW		RW				
Name																									BUFFERABLE	ILMODE		DSTILEN				

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7	BUFFERABLE	0x0	RW	**Allow AHB buffering**
	Determines whether destination writes can be a bufferable AHB transaction			
6:5	ILMODE	0x0	RW	**Interleave Mode**
	Rules table interleaving mode			

Value	Mode	Description
0	ABSOLUTE	Address determined by value in rules. Size of WORD
1	RELATIVE16	Address determined by adding rules to DST. Size of HALF-WORD
2	RELATIVE8	Address determined by adding rules to DST. Size of BYTE

4	DSTILEN	0x0	RW	**Destination Interleave**
	Enables destination interleaving feature			
3:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

26.7.33 LDMA_CHx_ILSRC - Interleaving Source Address

Offset	Bit Position																															
0x090	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	RW																															
Name	ADDR																															

Bit	Name	Reset	Access	Description
31:0	ADDR	0x0	RW	**Interleave Source Address**
	Base address for rules table in memory			

26.8 LDMAXBAR Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LDMAXBAR_IPVERSION	R	IP Veersion ID
0x004	LDMAXBAR_CHx_REQSEL	RW	Channel Peripheral Request Select Reg...
0x1000	LDMAXBAR_IPVERSION_SET	R	IP Veersion ID
0x1004	LDMAXBAR_CHx_REQSEL_SET	RW	Channel Peripheral Request Select Reg...
0x2000	LDMAXBAR_IPVERSION_CLR	R	IP Veersion ID
0x2004	LDMAXBAR_CHx_REQSEL_CLR	RW	Channel Peripheral Request Select Reg...
0x3000	LDMAXBAR_IPVERSION_TGL	R	IP Veersion ID
0x3004	LDMAXBAR_CHx_REQSEL_TGL	RW	Channel Peripheral Request Select Reg...

26.9 LDMAXBAR Register Description

26.9.1 LDMAXBAR_IPVERSION - IP Veersion ID

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x3																															
Access	R																															
Name	IPVERSION																															

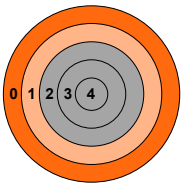
Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x3	R	**IP Version ID**
	ID indicating version of IP			

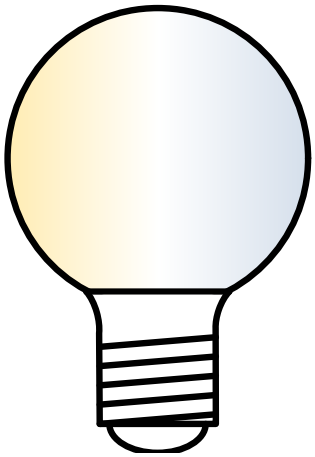
26.9.2 LDMAXBAR_CHx_REQSEL - Channel Peripheral Request Select Reg...

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0																		0x0		
Access											RW																		RW		
Name											SOURCESEL																		SIGSEL		

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:16	SOURCESEL	0x0	RW	**Source Select**
	Select input source to DMA channel.			
15:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3:0	SIGSEL	0x0	RW	**Signal Select**
	Select input signal to DMA channel.			

27. LED Pre-Driver (LEDDRV)





Quick Facts

What?

The LEDDRV peripheral drives up to two power FET gates to control LED current in AC- or DC-powered LED light bulbs.

Why?

LEDDRV can replace a dedicated LED driver IC with external FETs, saving cost and allowing for more flexibility for system updates.

How?

A dedicated charge pump, FET gate drivers, and analog feedback are used to implement switched-mode control of two LED chains. PWM signals routed through PRS provide mixing and dimming with little processor intervention.

27.1 Introduction

The LEDDRV block drives two power FETs to control the LED current in an AC line-powered LED light. It employs Boundary Conduction Mode (BCM) techniques to efficiently control the LED current. The source current, drain voltage, and AC supply are monitored to provide over-current protection, efficient switch timing, and in-system timing calibration. A built-in charge pump circuit boosts a nominal 3.3 V input to supply 6 V for the FET driver outputs. The circuit can also be used for efficient LED control with DC supplies.

27.2 Features

- Boundary Conduction Mode (BCM) operation
- Constant on-time control with programmable peak current
- Valley detection off-time control
- Charge pump supply to generate FET gate voltages
- PWM dimming and mixing
- Over-current protection
- Thermal foldback protection
- Direct drive and dual-drive modes
- Available in EM0 and EM1

27.3 Functional Description

A block diagram of the LEDDRV block is shown in Figure 27.1. LEDDRV Overview. Three dedicated pins are used for the charge pump supply (LEDVDD) and FET gate controls (LEDGATE0 and LEDGATE1). The block makes use of ABUS to route the VACSENSE, DRNSENSE0, DRNSENSE1, IPKSENSE0 and IPKSENSE1 signals through GPIO on port C and D. PWM control inputs are routed using the PRS hardware.

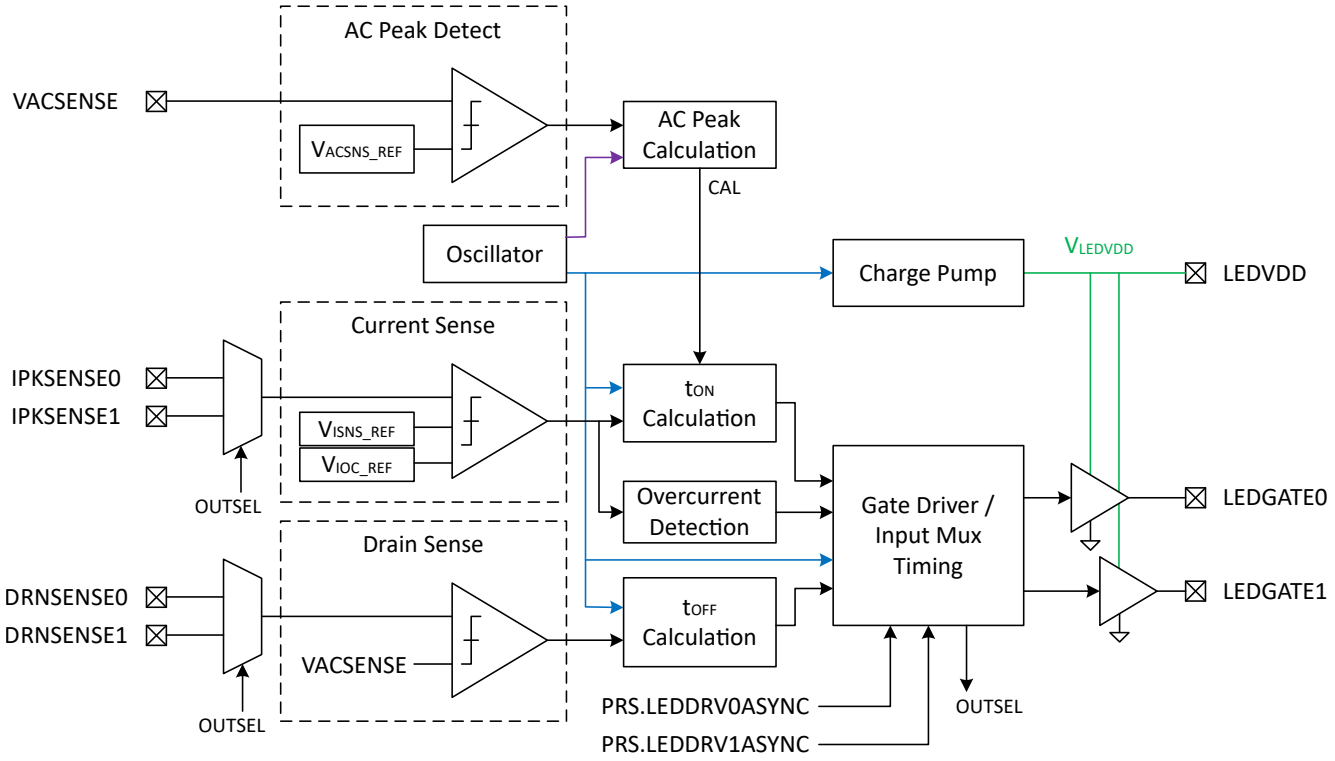


Figure 27.1. LEDDRV Overview

27.3.1 Normal BCM Operation

The normal mode of operation for the LEDDRV hardware is selected when CTRL.ENADUALMODE and CTRL.ENDIRDRV are both cleared to '0'. In this mode the circuitry implements a BCM control loop to control FET switching in a circuit such as the one shown in Figure 27.2. Example LED Schematic (One Channel).

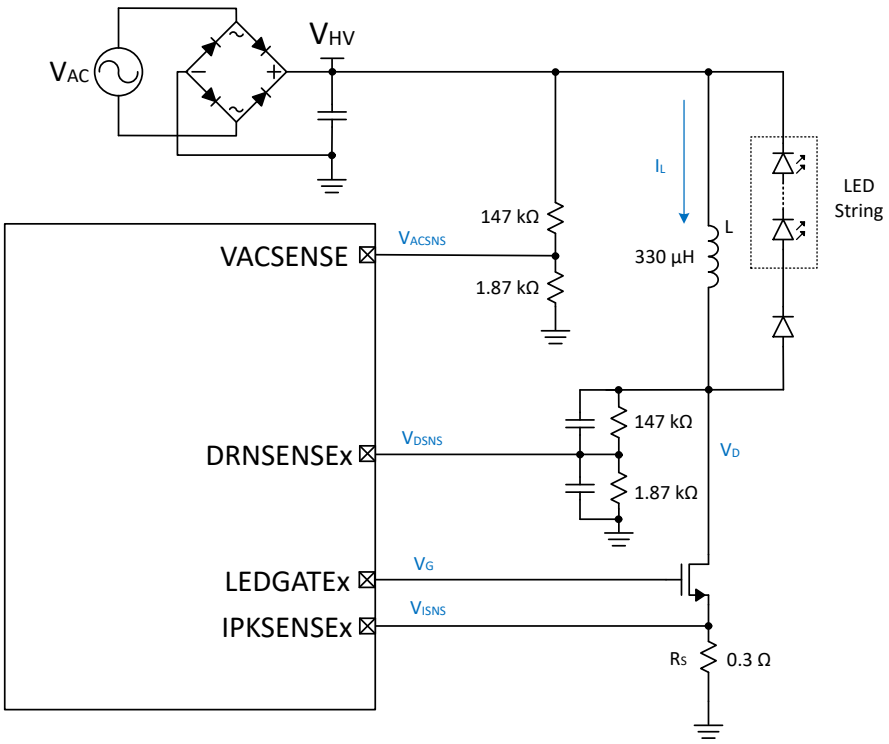


Figure 27.2. Example LED Schematic (One Channel)

Component values are examples, and may be adjusted to suit the needs of the application and the power supply characteristics.

The FET is switched on for a fixed time, t_{ON}, and the inductor current is ramped up. When the FET is switched off during the off time, t_{OFF}, the current in the inductor flows through the LED string. The inductor current is shown in Figure 27.3. Inductor Current.

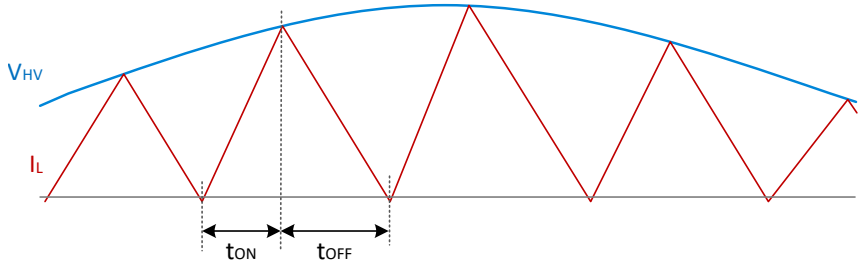


Figure 27.3. Inductor Current

Because t_{ON} is timed over a fixed interval, the peak current achieved during each switching cycle depends on the AC voltage.

$I_{peak} = (t_{ON} * V_{HV}(t))/ L$

The time t_{OFF} is dependent on both the peak current in the inductor and the time this takes to discharge through the LED string. The inductor current will reach zero, and then begin to go slightly negative. As a result, the drain voltage V_D will normally begin to ring. The voltage V_{DSNS} is monitored to determine when V_D hits a local minimum, and the off time is terminated to prevent the ringing.

Both the t_{ON} time and t_{OFF} time can be limited by hardware. The TONMAXVAL and TOFFMAXVAL fields in the LEDDRV_CFG register are used to specify maximum time, according to Table 27.1. Maximum t_{ON} and t_{OFF} Control.

Table 27.1. Maximum t_{ON} and t_{OFF} Control

TONMAXVAL or TOFFMAXVAL	Time
0	Disabled (no maximum time)
1	6.4 us
2	12.8 us
3	19.2 us
4	25.6 us
5	32.0 us
6	38.4 us
7	44.8 us

If TONMAXVAL is reached while the FET is on, the FET output is switched to the off state, even if the current t_{ON} has not been reached. Likewise, if TOFFMAXVAL is reached while the FET is off, the FET output will be switched to the on state. Note that these timeouts only apply when the channel is active. During inactive times (while the other channel is active, or both channels are inactive), the FET is switched off.

27.3.1.1 AC Peak Detection and t_{ON} Calibration

The LEDDRV block has hardware to determine where the peak AC voltage occurs, and to calibrate t_{ON} to a known peak current level. The voltage V_{ACSNS} is used to determine where AC peaks occur. V_{ACSNS} is compared against an internal reference threshold (V_{ACSNS_REF}), and a counter running from the LEDDRV oscillator determines how long the AC voltage stays above that threshold (t_{ACDET}). The resulting counter value is stored and on later AC cycles it is used to estimate the peak of the waveform, as shown in Figure 27.4. AC Peak Detection.

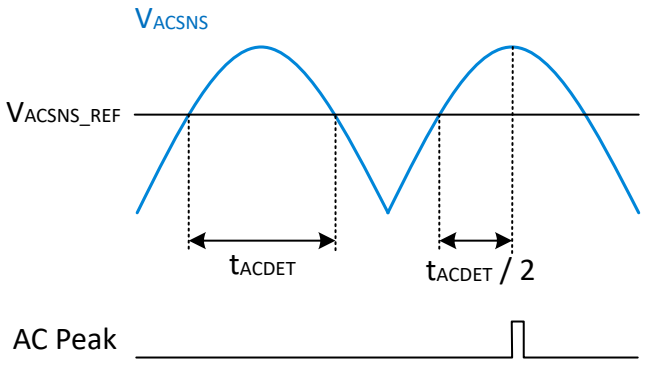


Figure 27.4. AC Peak Detection

The t_{ON} time can be calibrated against a peak current target (V_{ISNS_REF}). During a calibration, a counter running from the LEDDRV oscillator is started when LEDGATEx is switched high to turn the FET on. The inductor current for the channel is monitored at V_{ISNS} and compared against V_{ISNS_REF}. When V_{ISNS} reaches V_{ISNS_REF}, the LEDGATEx output is switched low to turn off the FET, and the counter value is stored. The counter value is then used to time subsequent t_{ON} events.

There are four options for calibrating t_{ON}, selected by the CFG.TONCALMODE bitfield.

- **ACTIVE** - The AC Peak detector logic is used to trigger a calibration near the AC voltage peak on the active channel. This is shown below in Figure 27.5. T_{ON} Calibration Using AC Peak.
- **BOTH** - The AC Peak detector logic is used to trigger a calibration near the AC voltage peak on both channels. The active channel is calibrated first, and then logic will briefly switch to the inactive channel and perform a calibration. Logic then switches back to the active channel.
- **FORCECAL** - Before FORCECAL is selected, firmware should set the CTRL.ENFRC bit to disable the AC Peak detector logic. A calibration will happen on the active channel when writing CTRL.TONFRCCAL from a 0 to a 1. CTRL.TONFRCCAL must transition from 0 to 1 to trigger a forced calibration.
- **FIXED** - t_{ON} time is not calibrated. The peak current detector logic is disabled. A fixed t_{ON} time is provided. The CFG.IPKDETREFADJ bits specify the 2 MSBs of the t_{ON} counter, and CFG.FIXEDTONSEL specifies the four LSBs of the t_{ON} counter. The t_{ON} counter is clocked from the local oscillator at a nominal frequency of 40 MHz.

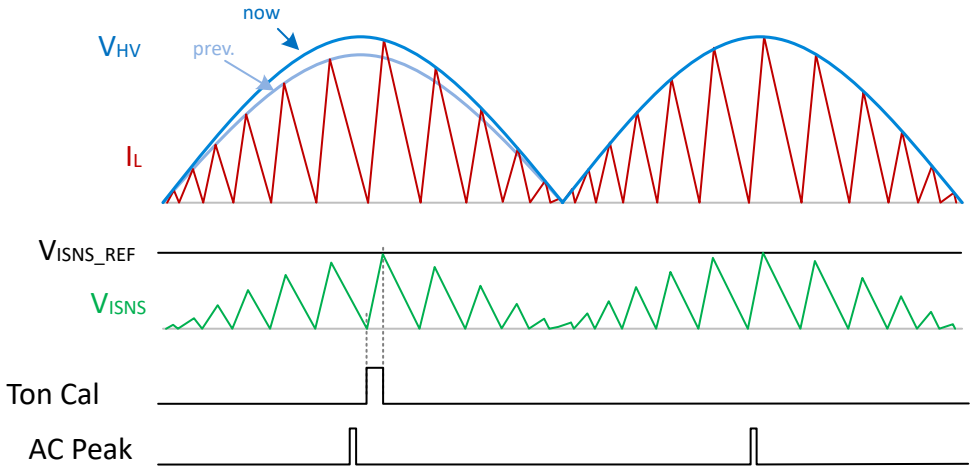


Figure 27.5. T_{ON} Calibration Using AC Peak

27.3.1.2 Peak Current Reference Control

The V_{ISNS_REF} voltage used for inductor peak current calibration is adjustable between four levels in 100 mV increments using the CFG.IPKDETREFADJ field, as shown in Table 27.2. Peak Current Reference Voltage.

Table 27.2. Peak Current Reference Voltage

IPKDETREFADJ	Nominal V_{ISNS_REF}
0	200 mV
1	300 mV
2	400 mV
3	500 mV

The inductor current these levels represent depend on the value of the current sense resistor as:

$I_L = V_{ISNS_REF} / R_S$

27.3.1.3 Over Current Protection

The LEDDRV block has over-current detection and protection capabilities. The inductor current is monitored for the active channel at the using the IPKSENSE input. The voltage at this pin (V_{ISNS}). The CFG.IOCREFADJ field is used to control the over-current threshold V_{IOC_REF}, in 200 mV increments as shown in Table 27.3. Over Current Reference Voltage.

Table 27.3. Over Current Reference Voltage

IOCREFADJ	Nominal V_{IOC_REF}
0	400 mV
1	600 mV
2	800 mV
3	1000 mV

The inductor current these levels represent depend on the value of the current sense resistor as:

$I_L = V_{IOC_REF} / R_S$

By default, the LEDDRV over-current hardware state machine is enabled, and monitors both channels. When V_{ISNS} exceeds V_{IOC_REF} on a channel during three successive cycles, the STATUS.OCFLAGx for that channel is set. The hardware state machine counts this as one event, and automatically clears the STATUS.OCFLAGx. If a second event occurs within 80,000 PCLK cycles the block will shut down for approximately 5 µs, then be re-activated.

The over-current hardware can be disabled in the case where software wants to handle over-current events directly. To disable the over-current hardware, set CTRL.HWOCFLAGDIS. The STATUS.OCFLAGx bit will be set when V_{ISNS} exceeds V_{IOC_REF} on a channel during three successive cycles. To clear this flag, software must write '1' to CTRL.SWCLROCFLAGx. The STATUS.SWCLROCFLAGSTSx bit will be set when a clear command is pending.

The IF.OCFLAG interrupt flag will be set if an over-current event is detected on either channel, and an interrupt will be generated if enabled.

27.3.1.4 Thermal Foldback and Over-Temperature Protection

To protect against over-temperature conditions, software can use the on-chip EMU temperature sensor (or an external sensor such as a thermistor via the ADC or ACMP) to monitor temperature. If temperature exceeds a desired threshold, software can disable the LEDDRV block entirely, or kill the PWM signals into the block to halt the drivers. Software may re-enable the block or PWM signals when conditions are appropriate.

Another option is to set the CTRL.IPKFOLDBACK bit to '1'. This does not completely kill the operation of the block. Instead, the t_{ON} time counter value is immediately cut in half, reducing the peak inductor current by half. The V_{ISNS_REF} is also halved so that any subsequent t_{ON} calibrations result in a similar peak current. When conditions are appropriate, CTRL.IPKFOLDBACK can be cleared to '0'. The t_{ON} time will be re-established at the next calibration cycle.

27.3.2 ABUS Signal Routing

The LEDDRV block requires three bus resources from the ABUS infrastructure to route the VACSENSE, IPKSENSE0/1 and DRNSENSE0/1 signals from pins on GPIO Ports C and D to the LEDDRV feedback circuitry. Some combinations routing these five signals to pins on port C and D will not be possible, given the constraints of ABUS.

Port C and Port D are served by a total of four ABUS buses: CDEVEN0, CDEVEN1, CDODD0, and CDODD1. The EVEN bus options support connection to any even-numbered pin, while the ODD bus options support connection to any odd-numbered pin. The IPKSENSE0 and IPKSENSE1 signals are not used simultaneously, and can share a single bus. Likewise, the DRNSENSE0 and DRNSENSE1 signals are not used simultaneously, and can share a single bus. The VACSENSE signal will use its own bus.

It is not possible to route an IPKSENSE signal, a DRNSENSE signal and the VACSENSE signal all to pins of the same polarity (EVEN or ODD) because there are not three EVEN or ODD buses available to the ports.

For simplicity, it is recommended to choose opposite bus polarities for the IPKSENSE and DRNSENSE signals. As an example, consider a system with IPKSENSE0 connected to PC01, IPKSENSE1 to PD01, DRNSENSE0 to PC00, DRNSENSE1 to PD00, and VACSENSE to PD02. The ABUS needs one ODD bus for the IPKSENSE signals, and two EVEN buses for VACSENSE and the DRNSENSE signals. Buses CDODD0, CDEVEN0, and CDEVEN1 should be allocated to LEDDRV.

27.3.3 Charge Pump Operation

The on-chip charge pump automatically starts when the LEDDRV module is enabled. A 100 nF bypass capacitor is required at the LEDVDD pin, which acts as the charge pump supply output. The charge pump will boost a nominal 3.3 V input supply at IOVDD to 6.0 V. This supply is used by the LEDGATE0 and LEDGATE1 outputs to drive power FET gates.

27.3.4 DC Supply Operation

In some lighting products, a high voltage DC supply is used to power the LEDs. LEDDRV can also be used in a DC system to implement BCM control of the LED currents. The circuit will operate as described in 27.3.1 Normal BCM Operation, except that the AC Peak detection will not be used. DC BCM control systems should use the fixed on-time (CFG.TONCALMODE = FIXED) or force calibration (CFG.TONCALMODE = FORCE) options.

27.3.5 Dimming and Mixing

Dimming and mixing control of the two channels is performed using two hardware PRS channel inputs: LEDDRV0ASYNC and LEDDRV1ASYNC. These may be connected to TIMER channel outputs to provide PWM signalling to the LEDDRV block. In the normal mode of operation, LEDDRV0ASYNC acts as a channel select input, while LEDDRV1ASYNC acts as a dimming input. Figure 27.6. PWM Control shows an example of how the mixing and dimming inputs enable each output.

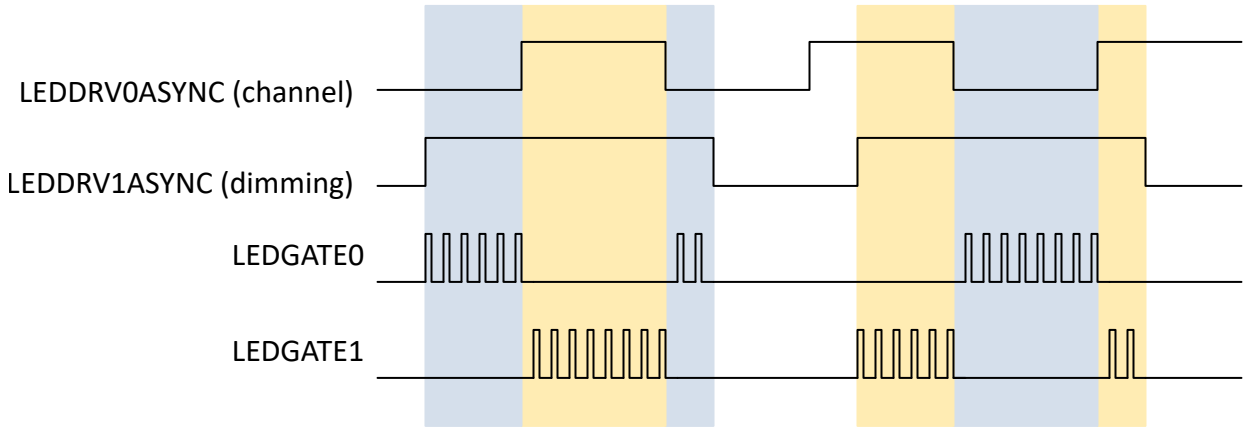


Figure 27.6. PWM Control

The LEDDRV0ASYNC channel select input can be used to provide color mixing for two LED strings. When LEDDRV0ASYNC is low, channel 0 will be active, and when LEDDRV0ASYNC is high, channel 1 will be active.

The LEDDRV1ASYNC dimming input can be used to provide overall dimming for both LED strings. When LEDDRV1ASYNC is high, the driver control logic will be active, and switching according to t_{ON} and t_{OFF}. When LEDDRV1ASYNC is low, the driver control logic will be off.

The driver control logic ensures that any cycle in progress will complete before disabling the driver and/or switching to the other channel.

27.3.6 Dual Driver Mode

In dual driver mode, the two output channels LEDGATE0 and LEDGATE1 act as a single output, and operate in phase with one another. The outputs are driven by the LEDDRV1ASYNC signal, while LEDDRV0ASYNC determines which feedback path to use. LEDDRV0ASYNC will typically be unassigned to a PRS channel when using dual drive mode, and so the channel 0 feedback is used. Dual drive mode is enabled by setting the CTRL.ENADUALMODE bit.

27.3.7 Direct Drive Mode

In direct drive mode, the control loop is disabled, and the LEDDRV block simply passes the inputs of LEDDRV0ASYNC and LEDDRV1ASYNC through to LEDGATE0 and LEDGATE1, respectively. Direct drive mode is enabled by setting the CTRL.ENDIRDRV bit.

27.4 LEDDRV Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	LEDDRV_IPVERSION	R	IP VERSION Register
0x004	LEDDRV_EN	RW ENABLE	Enable Register
0x008	LEDDRV_SWRST	RW SWRST	SWRST Register
0x014	LEDDRV_CFG	RW	Configuration Register
0x018	LEDDRV_CTRL	RW	Control Register
0x01C	LEDDRV_STATUS	RWH SYNC	Status Register
0x020	LEDDRV_ABUSCTRL0	RW CONFIG	LEDDRV ABUS Control 0
0x024	LEDDRV_ABUSCTRL1	RW CONFIG	LEDDRV ABUS Control 1
0x028	LEDDRV_IF	RWH INTFLAG	Interrupt Flag Register
0x02C	LEDDRV_IEN	RW	Interrupt Enable Register
0x1000	LEDDRV_IPVERSION_SET	R	IP VERSION Register
0x1004	LEDDRV_EN_SET	RW ENABLE	Enable Register
0x1008	LEDDRV_SWRST_SET	RW SWRST	SWRST Register
0x1014	LEDDRV_CFG_SET	RW	Configuration Register
0x1018	LEDDRV_CTRL_SET	RW	Control Register
0x101C	LEDDRV_STATUS_SET	RWH SYNC	Status Register
0x1020	LEDDRV_ABUSCTRL0_SET	RW CONFIG	LEDDRV ABUS Control 0
0x1024	LEDDRV_ABUSCTRL1_SET	RW CONFIG	LEDDRV ABUS Control 1
0x1028	LEDDRV_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x102C	LEDDRV_IEN_SET	RW	Interrupt Enable Register
0x2000	LEDDRV_IPVERSION_CLR	R	IP VERSION Register
0x2004	LEDDRV_EN_CLR	RW ENABLE	Enable Register
0x2008	LEDDRV_SWRST_CLR	RW SWRST	SWRST Register
0x2014	LEDDRV_CFG_CLR	RW	Configuration Register
0x2018	LEDDRV_CTRL_CLR	RW	Control Register
0x201C	LEDDRV_STATUS_CLR	RWH SYNC	Status Register
0x2020	LEDDRV_ABUSCTRL0_CLR	RW CONFIG	LEDDRV ABUS Control 0
0x2024	LEDDRV_ABUSCTRL1_CLR	RW CONFIG	LEDDRV ABUS Control 1
0x2028	LEDDRV_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x202C	LEDDRV_IEN_CLR	RW	Interrupt Enable Register
0x3000	LEDDRV_IPVERSION_TGL	R	IP VERSION Register
0x3004	LEDDRV_EN_TGL	RW ENABLE	Enable Register
0x3008	LEDDRV_SWRST_TGL	RW SWRST	SWRST Register
0x3014	LEDDRV_CFG_TGL	RW	Configuration Register
0x3018	LEDDRV_CTRL_TGL	RW	Control Register
0x301C	LEDDRV_STATUS_TGL	RWH SYNC	Status Register
0x3020	LEDDRV_ABUSCTRL0_TGL	RW CONFIG	LEDDRV ABUS Control 0
0x3024	LEDDRV_ABUSCTRL1_TGL	RW CONFIG	LEDDRV ABUS Control 1

Offset	Name	Type	Description
0x3028	LEDDRV_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x302C	LEDDRV_IEN_TGL	RW	Interrupt Enable Register

27.5 LEDDRV Register Description

27.5.1 LEDDRV_IPVERSION - IP VERSION Register

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP version register**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

27.5.2 LEDDRV_EN - Enable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status**
	When EN is cleared, DISABLING is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFOs.			
0	EN	0x0	RW	**Module enable**
	The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.			

27.5.3 LEDDRV_SWRST - SWRST Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Reset busy status** When SWRST command is issued, resetting logic sets RESETTING status immediately, and later it is cleared when reset process finishes.
0	SWRST	0x0	W	**Software Reset bit** A software reset command field resets the module back to the initial condition, similar to a power on reset condition.

27.5.4 LEDDRV_CFG - Configuration Register

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0				0x1		0x3		0x1		0x7			0x7		
Access																	RW				RW		RW		RW		RW			RW		
Name																	FIXEDTONSEL				TONCALMODE		IOCREFADJ		IPKDETREFADJ		TOFFMAXVAL			TONMAXVAL		

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:12	FIXEDTONSEL	0x0	RW	**Fixed Ton Select LSBs**
	Sets four LSBs of Ton when TONCALMODE = FIXED. MSBs are taken from IPKDETREFADJ.			
11:10	TONCALMODE	0x1	RW	**Ton Calibration Mode Select**
	Ton calibration mode selection			

Value	Mode	Description
0	ACTIVE	Calibrates only the active channel automatically
1	BOTH	Calibrates both channels automatically
2	FORCECAL	Force calibration on active channel when TONFRCCAL is set
3	FIXED	Use fixed Ton settings (no calibration)

9:8	IOCREFADJ	0x3	RW	**Overcurrent Reference**
	Over-current reference selection			
7:6	IPKDETREFADJ	0x1	RW	**Peak Current Reference**
	Peak current reference selection when TONCALMODE != FIXED. Two MSBs of Ton when TONCALMODE = FIXED.			
5:3	TOFFMAXVAL	0x7	RW	**Toff Maximum Value**
	Sets upper limit for Toff			
2:0	TONMAXVAL	0x7	RW	**Ton Maximum Value**
	Sets upper limit for Ton			

27.5.5 LEDDRV_CTRL - Control Register

Offset	Bit Position																													
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																				0x0	0x0	0x0						0x0	0x0	0x0	0x0	0x0
Access																				RW	RW	RW						RW	RW	RW	RW	RW
Name																				SWCLROCFLAG1	SWCLROCFLAG0	HWOCFLAGDIS						ENFRC	TONFRCCAL	ENDIRDRV	IPKFOLDBACK	ENADUALMODE

Bit	Name	Reset	Access	Description
31:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	SWCLROCFLAG1	0x0	RW	**Software Clear OCFLAG channel 1**
	Clear OCFLAG1 events (when HWOCFLAGDIS = 1)			
11	SWCLROCFLAG0	0x0	RW	**Software Clear OCFLAG channel 0**
	Clear OCFLAG0 events (when HWOCFLAGDIS = 1)			
10	HWOCFLAGDIS	0x0	RW	**Disable Hardware Over current handling**
	Disables the hardware controlled OCFLAG state machine to provide software control over OCFLAG event handling			
9:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	ENFRC	0x0	RW	**Enable Force calibration**
	Enables forced calibration option. Disables the internal enable that is gated on AC peak detect			
3	TONFRCCAL	0x0	RW	**Ton Force calibration**
	Force calibration on positive edge when TONCALMODE = FORCECAL			
2	ENDIRDRV	0x0	RW	**Direct Drive Enable**
	When set to 1, enables direct drive mode. The two PWM inputs will be applied directly to the gates.			
1	IPKFOLDBACK	0x0	RW	**Peak current foldback enable**
	When set to 1, the Ton setting is immediately divided by two to reduce current flow in the LED driver, and the calibration reference is also cut in half.			
0	ENADUALMODE	0x0	RW	**Dual driver mode enable**
	When set to 1, enables parallel drive mode (both channels driven at the same time). The channel selection input is ignored.			

27.5.6 LEDDRV_STATUS - Status Register

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																		0x0	0x0	0x0	0x0	0x0	0x0	0x0				0x0				
Access																		RW	RW	R	R	RW	RW	R				R				
Name																		SWCLROCFLAGSTS1	SWCLROCFLAGSTS0	ABUSINPUTCONFLICT	ABUSALLOCERR	OCFLAG1	OCFLAG0	VALDETCNT				ACDETCNT				

Bit	Name	Reset	Access	Description
31:15	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
14	SWCLROCFLAGSTS1	0x0	RW	**Software OCFLAG1 Clear Pending**
	Set when CTRL.SWCLROCFLAG1 is used to clear the OCFLAG1. This bit will be cleared after hardware clears OCFLAG1.			
13	SWCLROCFLAGSTS0	0x0	RW	**Software OCFLAG0 Clear Pending**
	Set when CTRL.SWCLROCFLAG0 is used to clear the OCFLAG0. This bit will be cleared after hardware clears OCFLAG0.			
12	ABUSINPUTCONFLICT	0x0	R	**ABUS Input Conflict Status**
	Set to 1 if VACSENSE or IPKSENSEx or DRNSENSEx request the same ABUS.			
11	ABUSALLOCERR	0x0	R	**ABUS Allocation Error Status**
	Set to 1 if a requested ABUS is not allocated to LEDDRV.			
10	OCFLAG1	0x0	RW	**Over Current Channel 1**
	Set to 1 if overcurrent event was detected on channel 1.			
9	OCFLAG0	0x0	RW	**Over Current Channel 0**
	Set to 1 if overcurrent event was detected on channel 0.			
8:5	VALDETCNT	0x0	R	**Valley detect counter output**
	Holds valley detection counter value			
4:0	ACDETCNT	0x0	R	**AC peak detect count value**
	Holds AC peak detection counter value			

27.5.7 LEDDRV_ABUSCTRL0 - LEDDRV ABUS Control 0

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0						0x0	
Access																									RW						RW	
Name																									VACPINSEL						VACPORTSEL	

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:4	VACPINSEL	0x0	RW	**VACSENSE Pin Select** Selects GPIO port pin number for VACSENSE
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	VACPORTSEL	0x0	RW	**VACSENSE Port Select** Selects GPIO Port for VACSENSE

Value	Mode	Description
0	NONE	VACSENSE not routed to ABUS
1	PORTC	Route to Port C (Select pin with VACPINSEL)
2	PORTD	Route to Port D (Select pin with VACPINSEL)

27.5.8 LEDDRV_ABUSCTRL1 - LEDDRV ABUS Control 1

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0						0x0		0x0						0x0		0x0						0x0		0x0						0x0	
Access	RW						RW		RW						RW		RW						RW		RW						RW	
Name	DSNSPINSEL1						DSNSPORTSEL1		ISNSPINSEL1						ISNSPORTSEL1		DSNSPINSEL0						DSNSPORTSEL0		ISNSPINSEL0						ISNSPORTSEL0	

Bit	Name	Reset	Access	Description
31:28	DSNSPINSEL1	0x0	RW	**DRNSENSE1 Pin Select**
	Selects GPIO port pin number for DRNSENSE1			
27:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:24	DSNSPORTSEL1	0x0	RW	**DRNSENSE1 Port Select**
	Selects GPIO Port for DRNSENSE1			

Value	Mode	Description
0	NONE	DRNSENSE1 not routed to ABUS
1	PORTC	Route to Port C (Select pin with DSNSPINSEL1)
2	PORTD	Route to Port D (Select pin with DSNSPINSEL1)

Bit	Name	Reset	Access	Description
23:20	ISNSPINSEL1	0x0	RW	**IPKSENSE1 Pin Select**
	Selects GPIO port pin number for IPKSENSE1			
19:18	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
17:16	ISNSPORTSEL1	0x0	RW	**IPKSENSE1 Port Select**
	Selects GPIO Port for IPKSENSE1			

Value	Mode	Description
0	NONE	IPKSENSE1 not routed to ABUS
1	PORTC	Route to Port C (Select pin with ISNSPINSEL1)
2	PORTD	Route to Port D (Select pin with ISNSPINSEL1)

Bit	Name	Reset	Access	Description
15:12	DSNSPINSEL0	0x0	RW	**DRNSENSE0 Pin Select**
	Selects GPIO port pin number for DRNSENSE0			
11:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9:8	DSNSPORTSEL0	0x0	RW	**DRNSENSE0 Port Select**
	Selects GPIO Port for DRNSENSE0			

Value	Mode	Description
0	NONE	DRNSENSE0 not routed to ABUS
1	PORTC	Route to Port C (Select pin with DSNSPINSEL0)

Bit	Name	Reset	Access	Description
	2	PORTD		Route to Port D (Select pin with DSNSPINSEL0)
7:4	ISNSPINSEL0	0x0	RW	**IPKSENSE0 Pin Select**
	Selects GPIO port pin number for IPKSENSE0			
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1:0	ISNSPORTSEL0	0x0	RW	**IPKSENSE0 Port Select**
	Selects GPIO Port for IPKSENSE0			

Value	Mode	Description
0	NONE	IPKSENSE0 not routed to ABUS
1	PORTC	Route to Port C (Select pin with ISNSPINSEL0)
2	PORTD	Route to Port D (Select pin with ISNSPINSEL0)

27.5.9 LEDDRV_IF - Interrupt Flag Register

Offset	Bit Position																														
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														OCFLAG	ABUSINPUTCONFLICT	ABUSALLOCERR

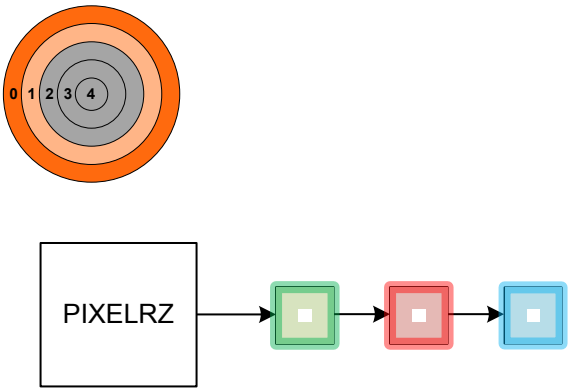
Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	OCFLAG	0x0	RW	**OCFLAG Interrupt Flag**
	Set if over-current condition is detected			
1	ABUSINPUTCONFLICT	0x0	RW	**ABUS Input Conflict Error Flag**
	Set if VACSENSE or IPKSENSEx or DRNSENSEx request the same ABUS. Should only be enabled when using ABUS for LEDDRV feedback.			
0	ABUSALLOCERR	0x0	RW	**ABUS Port Allocation Error Flag**
	Set if a reqeusted ABUS is not allocated to LEDDRV. Should only be enabled when using ABUS for LEDDRV feedback.			

27.5.10 LEDDRV_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														RW	RW	RW
Name																														OCFLAG	ABUSINPUTCONFLICT	ABUSALLOCERR

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	OCFLAG	0x0	RW	**OCFLAG Interrupt Enable**
	Set to enable the OCFLAG Interrupt			
1	ABUSINPUTCONFLICT	0x0	RW	**ABUS Input Conflict Interrupt Enable**
	Set to enable the ABUSINPUTCONFLICT Interrupt			
0	ABUSALLOCERR	0x0	RW	**ABUS Allocation Error Interrupt Enable**
	Set to enable the ABUSALLOCERR Interrupt			

28. PIXELRZ - Serial Pixel Interface



Quick Facts

What?

The PIXELRZ sends an RZ serial bitstream on a single wire output to communicate to addressable LED controller ICs.

Why?

Hardware control of the bus timing and data formatting allows communication to a large number of LED controller ICs with minimal processor intervention.

How?

Dedicated hardware uses DMA and configurable symbol timing to send frames of data to the external LED controller chips.

28.1 Introduction

PIXELRZ is an interface peripheral that transmits data to an external LED driver network using return-to-zero encoding protocol. Many commercially available LED pixel driver ICs use this protocol. Data bits and special characters are encoded as symbols with a specified high and low time. Every symbol returns to zero after a certain period of being high (set to 1). The PIXELRZ peripheral makes use of the DMA to pull frames from memory, unpacks the data, and sends it serially to the LED drivers without CPU intervention.

28.2 Features

- Supports sequential and parallel bus topologies
- Independent timing control of t_{HIGH} and t_{LOW} periods for all symbols
- 8-deep 32-bit wide FIFO
- Supports 8-bit or 32-bit data alignment
- DMA support with configurable watermark level
- Output inversion
- Software or hardware PRS trigger options
- Hardware jitter

28.3 Functional Description

An overview of the PIXELRZ block is shown in Figure 28.1. PIXELRZ Block Diagram.

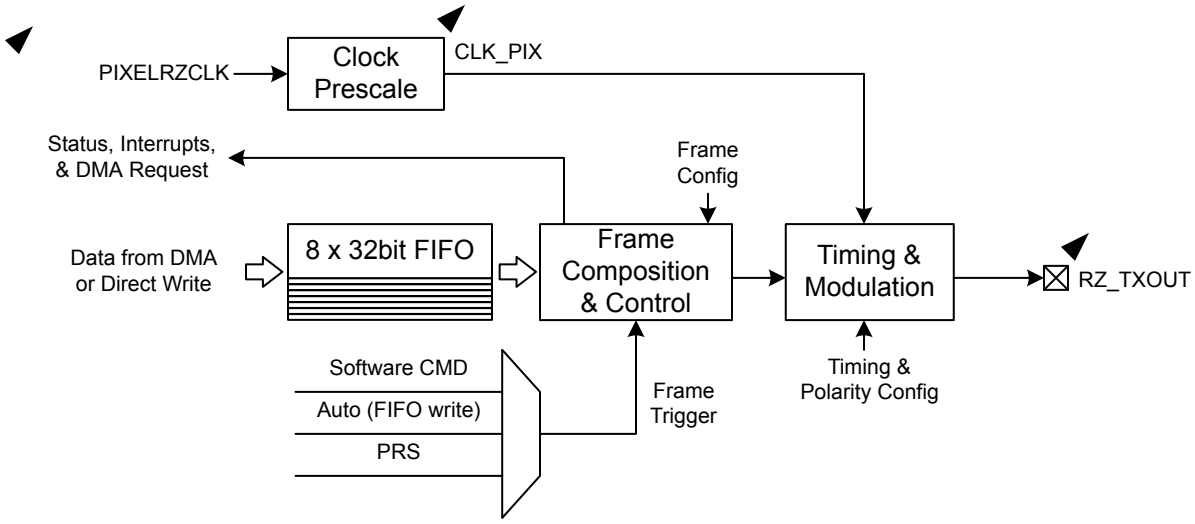


Figure 28.1. PIXELRZ Block Diagram

The PIXELRZ includes a transmit FIFO and the necessary hardware to format and serialize incoming data into frames. A flexible timing control block provides modulation for different bus symbols. The PIXELRZ interface operates down to EM1.

28.3.1 Sequential Mode Operation

A sequentially-connected LED bus example is shown in Figure 28.2. Sequential LED Connections.

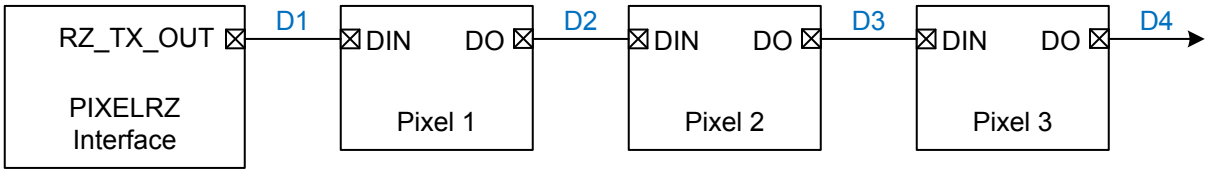


Figure 28.2. Sequential LED Connections

The PIXELRZ block is configured to operate in sequential mode by setting the CFG.MODE field to SEQUENTIAL.When operating in sequential mode, CFG.RESETEN should be set to ENABLE to enable RESET symbol generation. The state of CFG.STOPEN is ignored in sequential mode.

In a sequential connected system, the bus controller (PIXELRZ) sends data in frames, separated by a RESET symbol. A single frame consists of the data for all of the pixels on the bus in order, followed by optional trailing data, as shown in Figure 28.3. Sequential Data Transmission.

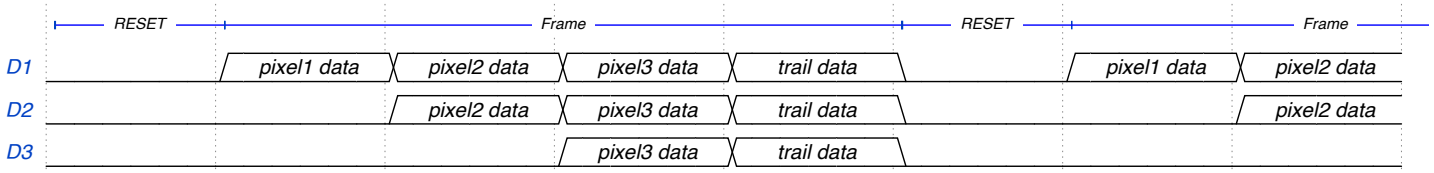


Figure 28.3. Sequential Data Transmission

Each pixel in the chain has an LED controller IC with a serial data input (DIN) and a serial data output (DO). The pixel ICs receive data on DIN, and store the first block of data to their own settings. All data blocks after the first block are mirrored on DOUT and sent to the next pixel in the chain. Thus, Pixel1 sees all data for Pixel1 through PixelN, Pixel2 sees data only for Pixel2 through PixelN, and so on. The last IC in the chain, PixelN, will only receive data for PixelN.

Some pixel ICs also support an additional global data field at the end of the pixel data, shown as 'trail data' in the figure. This data is received and stored for all pixels in the chain. The RESET symbol at the end of the stream signals to all pixels that they should apply their data and any trail data, allowing the changes to be synchronous across all pixels in the chain.

28.3.1.1 Sequential Symbols

In a typical sequential bus configuration, only the DATA0, DATA1 and RESET symbols are used. The DATA0 and DATA1 symbols are often the same width, differentiated by the amount of time the data line remains high. RESET symbols are typically specified as a minimum time the bus is held low, with no high period, and much longer than either DATA symbol. This is illustrated in Figure 28.4. Typical Sequential Symbols.

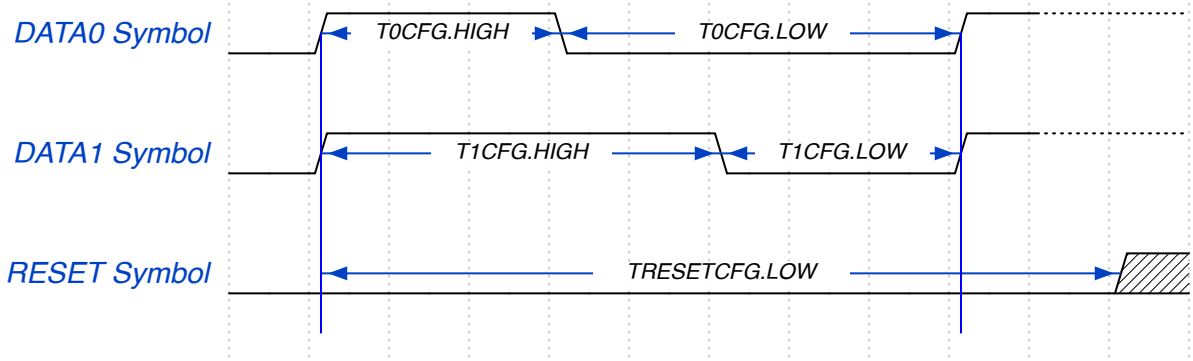


Figure 28.4. Typical Sequential Symbols

28.3.2 Parallel Mode Operation

A parallel-connected LED bus example is shown in Figure 28.5. Parallel LED Connections.

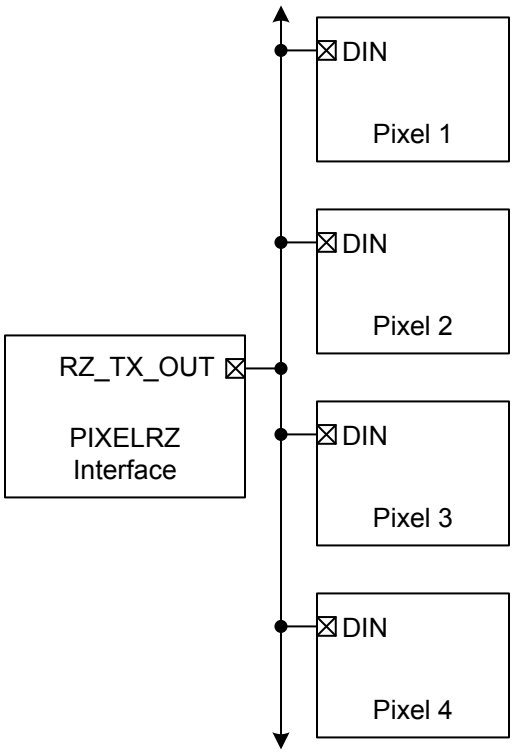


Figure 28.5. Parallel LED Connections

The PIXELRZ block is configured to operate in parallel mode by setting the CFG.MODE field to PARALLEL. When operating in parallel mode, CFG.STOPEN should be set to ENABLE to enable STOP symbol generation. The state of CFG.RESETEN is ignored in parallel mode.

In a parallel connected system, the data output line from the bus controller (PIXELRZ) runs to the input of all the devices on the bus. The pixels use random addressing, meaning that pixel data is sent on the bus including an address to designate the target pixel. Each packet of pixel data is followed by a STOP signal.

The PIXELRZ interface can organize the pixel data in frames, sending data to many different pixels on the bus in a single frame as shown in Figure 28.6. Parallel Data Transmission.

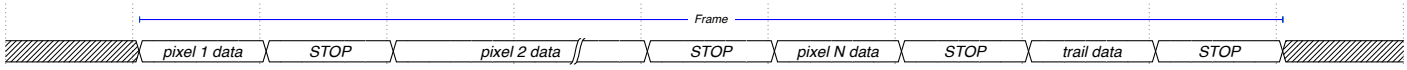


Figure 28.6. Parallel Data Transmission

Pixels may be addressed in any order, and not every pixel on the bus needs to be addressed in every frame. The PIXELRZ interface also supports sending trail data in parallel mode. Trail data is sent after all other pixels in a frame, followed by a STOP symbol.

28.3.2.1 Parallel Symbols

In a typical parallel bus configuration, only the DATA0, DATA1 and STOP symbols are used. The DATA0, DATA1, and STOP symbols are differentiated by the amount of time the data line remains high. Low time between symbols may only have a minimum timing requirement that is common for all symbols. This is illustrated in Figure 28.7. Typical Parallel Symbols.

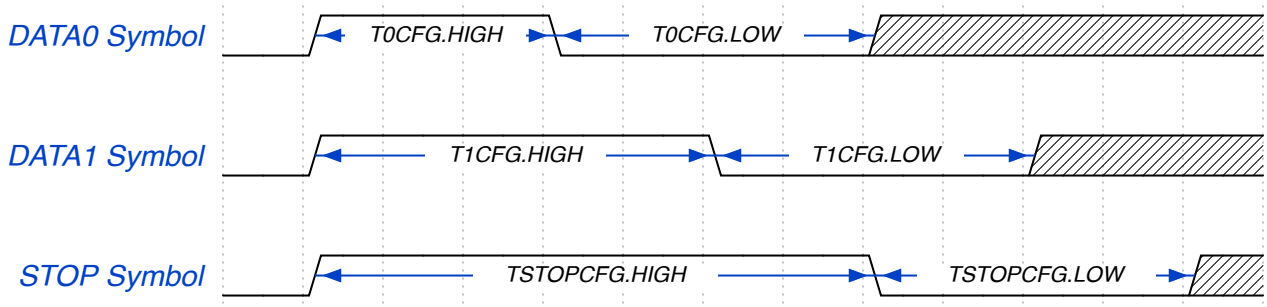


Figure 28.7. Typical Parallel Symbols

28.3.2.2 Spread Spectrum (Jitter)

Parallel pixel configurations use a single long data line, which can result in undesirable radiated emissions. To mitigate this, the PIXELRZ interface includes a feature to generate randomized timing jitter on the low cycle of DATA0 and DATA1 symbols. When CFG.JITTEREN is set to ENABLE, the interface will add a random amount of time between 1 and 16 CLK_PIX clocks to the LOW periods of T0 and T1 of every transmitted bit. The jitter is randomized using an internal linear feedback shift register, which produces an even distribution of values.

28.3.3 Clocking

The PIXELRZ block(s) are clocked from the PIXELRZCLK clock branch that is configured in the CMU. The input clock determines the accuracy and resolution of the resulting bit stream. PIXELRZCLK has an optional prescaler in the CMU to divide the selected oscillator source. The maximum clock rate for PIXELRZCLK is 40 MHz. PIXELRZCLK may be further divided inside of the PIXELRZ interface using the CFG.DIV field, to provide CLK_PIX, which is used for all symbol timing.

28.3.4 Symbol Generation

The PIXELRZ interface has registers to define the high and low timing for four different bus symbols: DATA0, DATA1, STOP, and RESET.

The timing for DATA0 and DATA1 symbols is controlled by the T0CFG and T1CFG registers respectively. Each of these registers has two 11-bit fields to specify the high and low periods of the symbol. Each period of these symbols can be configured to last for anywhere from 1 to 2047 cycles of CLK_PIX. A zero in the timing fields for T0 and T1 is not considered valid, and will result in one cycle for that period.

The STOP symbol timing is controlled by the TSTOPCFG register, with two 11-bit fields to specify the high and low timing. For the STOP symbol a zero is valid, and both the high and low time may be set between 0 and 2047 cycles of CLK_PIX.

The RESET symbol timing is controlled by the TRSTCFG register, with two 16-bit fields to specify the high and low timing periods. Each period of the RESET symbol can be configured to last for 0 to 65535 cycles of CLK_PIX. In most applications RESET will be specified only with a low time, and TRSTCFG.HIGH will be set to 0.

28.3.4.1 Output Inversion

The CFG.OUTINV bit field can be used to invert the bus output. When OUTINV is set to INVERT, all bus signalling is the opposite polarity from what is described in this document. Low times become high times, and vice-versa.

28.3.5 Frame Triggering

All PIXELRZ frames are sent on a trigger event, controlled by the CFG.TRIGSEL field. There are three options available as trigger sources.

1. **Software trigger**: When configured for software trigger mode, each frame requires that software set the CMD.TXEN bit to start the transfer. TXEN enables the transmission frame-by-frame. If there is data available in the FIFO when TXEN is set, the frame will begin immediately. If no data is present in the FIFO when TXEN is set, the peripheral will begin sending the frame when the first data is written to the FIFO. At the end of a frame, the peripheral will not initiate a new frame until software sets TXEN again.
2. **Automatic trigger**: When configured for automatic trigger mode, a new frame will be initiated whenever data is available in the FIFO. The TXEN command enables this mode. If there is data available in the FIFO when TXEN is set, the frame will begin immediately. If no data is present in the FIFO when TXEN is set, the peripheral will begin sending the frame when the first data is written to the FIFO. At the end of a frame, the peripheral will return to waiting on data in the FIFO to send the next frame. If the FIFO already contains more data when a frame completes, the next frame will start automatically.
3. **PRS trigger**: When configured for PRS trigger mode, the peripheral acts like software trigger mode to initiate the first frame. It will begin sending the first frame when CMD.TXEN has been set, and data is present in the FIFO. At the end of a frame, the peripheral will wait until recieving a PRS trigger to send the next frame. Like the other two modes, if a PRS trigger occurs and there is no data in the FIFO, the peripheral will wait until data has been written to start the next frame. To effectively use PRS trigger mode for precise frame timing, data for the next frame should be written to the FIFO before th PRS trigger occurs.

In the event that a frame transmission is in progress and needs to be stopped for any reason, the PIXELRZ interface provides two options. Setting the CMD.TXDIS bit to '1' will immediately halt the current transmission and reset the peripheral, removing any pending data from the FIFO. Setting CMD.CLEARTX to '1' will finish any pixel in progress before resetting the peripheral and clearing the FIFO.

28.3.6 Frame Formatting

The FRAMECFG register is used to define the parameters for a full frame. FRAMECFG.PIXELWIDTH defines the number of data bits per pixel from 1 to 1023. FRAMECFG.PIXELNUM defines the number of pixels in each frame, from 1 to 1023. FRAMECFG.TRAILWIDTH defines the number of bits in the optional trailing data, from 0 to 511.

If PIXELWIDTH or PIXELNUM are set to 0, this is treated as a special case in which the interface will continue to send out bits or pixel data indefinitely, as long as there is data available in the FIFO. The interface will stop sending data when a FIFO underflow occurs. The FIFO underflow will result in an error, and no trail data will be sent (even if specified by TRAILWIDTH). The RZ_TXOUT line will go low and remain low until the next frame is triggered. Note that in sequential mode, the RESET symbol is not explicitly applied to the bus in this case.

28.3.7 FIFO Interface

The PIXELRZ peripheral contains an 8-deep, 32-bit wide transmit FIFO to manage data. Software or DMA can write information into the FIFO using the TXDATA register, and the PIXELRZ state machine will pull the data out in order to transmit. The number of words currently in the FIFO waiting to be sent may be read at any time from STATUS.TXFCNT. The STATUS.TXFULL bit will indicate when the FIFO is completely full and cannot accept more data.

The bit field CFG.TXFWL is an adjustable watermark that configures when hardware will request more data. If the number of free slots in the FIFO is greater than CFG.TXFWL, the STATUS.TXFL flag is set to '1' to indicate that the watermark is crossed. This event can trigger a processor interrupt or a DMA request.

The FIFO logic can generate four different interrupts for the system. These are enabled in the IEN register, with interrupt flags in the IF register. The interrupt events will always set the associated IF register flag, and a processor interrupt will occur if the associated bit is set in the IEN register.

1. TXFL is the FIFO watermark level interrupt. It occurs when a word is popped from the FIFO and the number of entries falls below the designated watermark in CFG.TXFWL.
2. TXC is the transmission complete interrupt. It occurs after a frame has been sent on the bus. If the trigger mode is set to AUTO or PRS, and there are no new entries in the FIFO, this interrupt will be set. If the trigger mode is configured for software triggers, this bit is always set at the end of a frame, regardless if there is more data available.
3. TXOF is the FIFO overflow interrupt. This will occur if the FIFO is written when it is already full. A FIFO overflow will result in the written data being lost.
4. TXUF is the FIFO underflow interrupt. This will occur if the PIXELRF state machine tries to read from an empty FIFO. This event causes the frame transmission to be terminated.

28.3.8 Data Alignment and Bit Order

The PIXELRZ interface has two options for data alignment. Data written to the FIFO may be alilgned at 32-bit boundaries or 8-bit boundaries. Using 8-bit boundaries may provide more efficient packing of data in RAM. CFG.MEMALIGN defines how the PIXELRZ interface will interpret data alignment in the FIFO. When MEMALIGN is set to ALIGN8 the peripheral will expect all peripheral and trail data to begin at 8-bit boundaries in the FIFO. Likewise, if MEMALIGN is set to ALIGN32, then all pixels and trail data are expected to be aligned at 32-bit boundaries.

Data may also be sent on the bus MSB-first or LSB-first, which is controlled by CFG.MSBF. When MSBF is set to LSB the first bit sent on the bus for each block of pixel data will be the LSB, starting at the designated alignment boundary. When MSBF is set to MSB, the first bit on the bus is the MSB, starting at the alignment boundary.

The following figures depict the four possible combinations of aligmnent and bit order, and show how the data will be sent on the bus. All examples show bus data in sequential mode, sending two consecutive frames with the following settings:

- FRAMECFG.PIXELNUM = 2 (2 pixels)
- FRAMECFG.PIXELWIDTH = 38 (38 bits per pixel)
- FRAMECFG.TRAILWIDTH = 26 (26 bits of trail data)

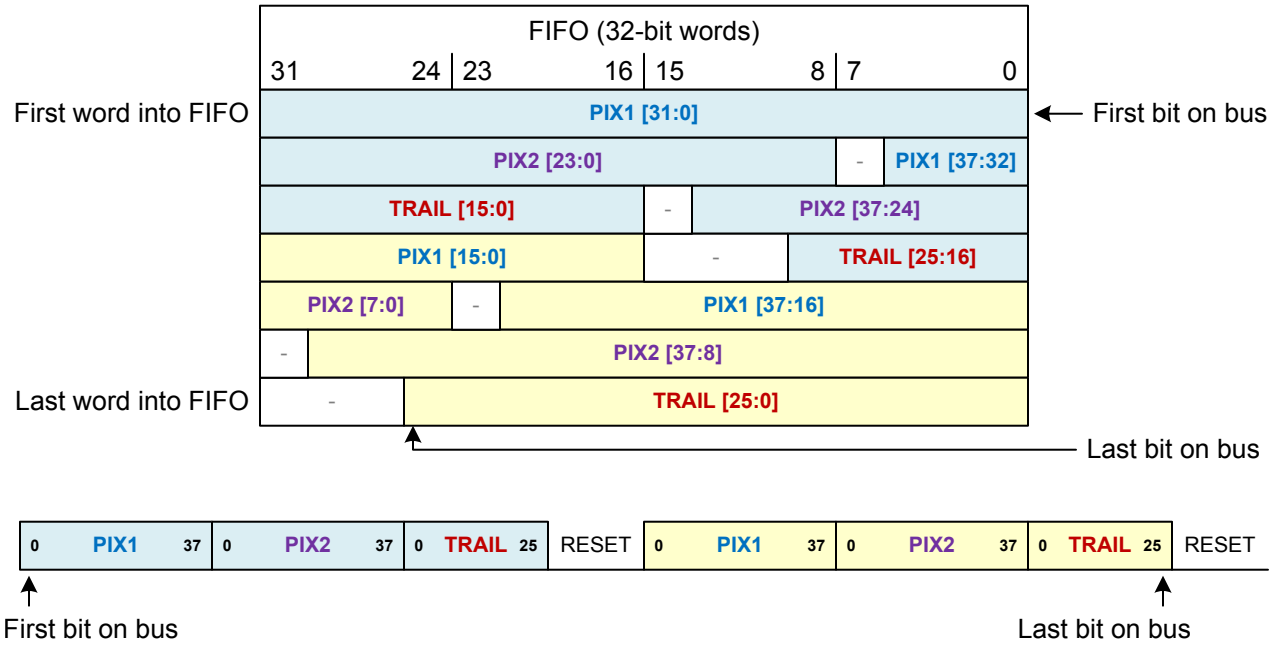


Figure 28.8. LSB-first with 8-bit Memory Alignment

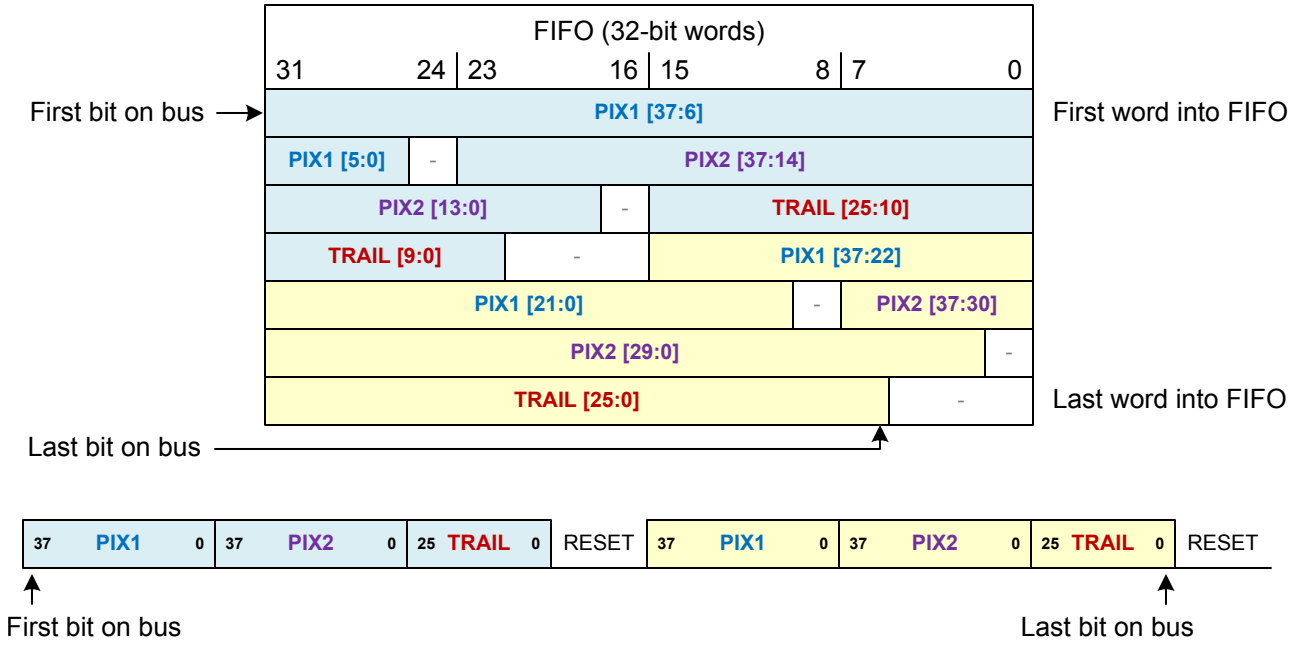


Figure 28.9. MSB-first with 8-bit Memory Alignment

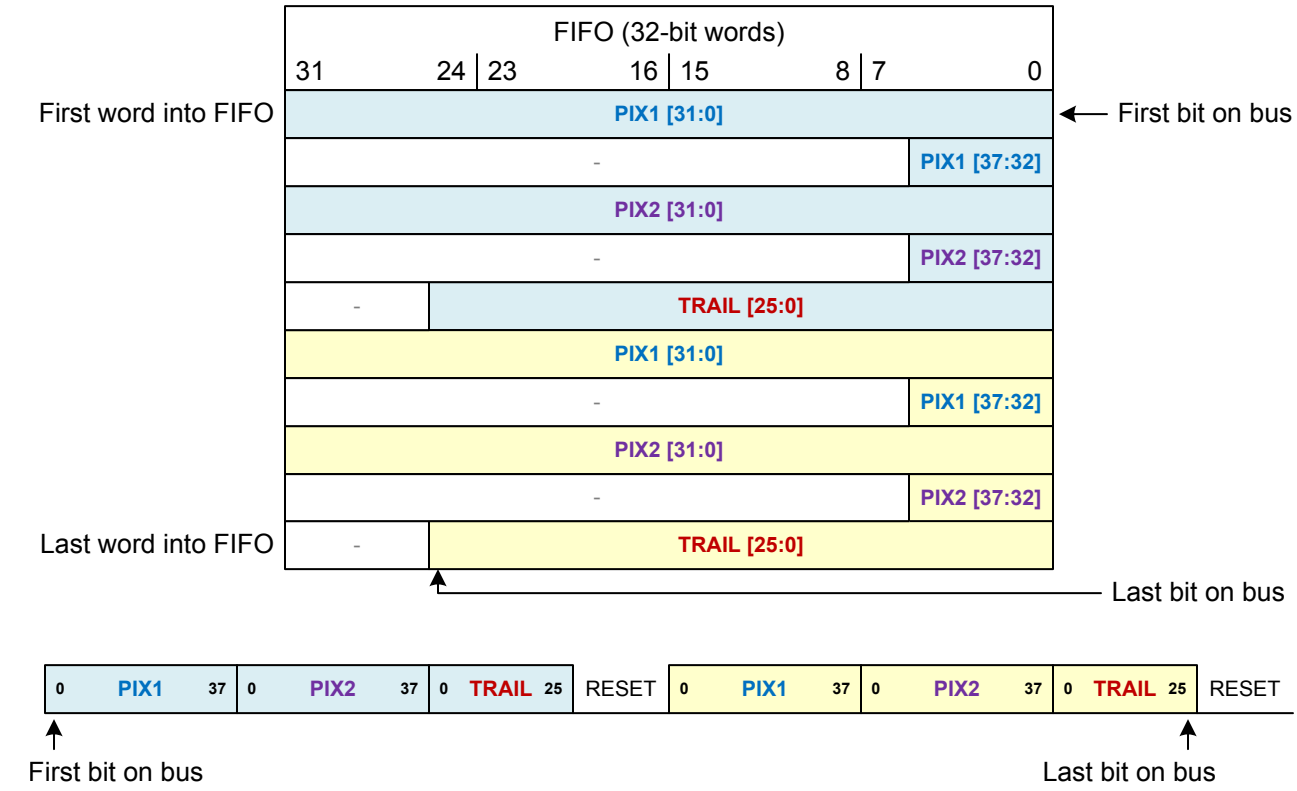


Figure 28.10. LSB-first with 32-bit Memory Alignment

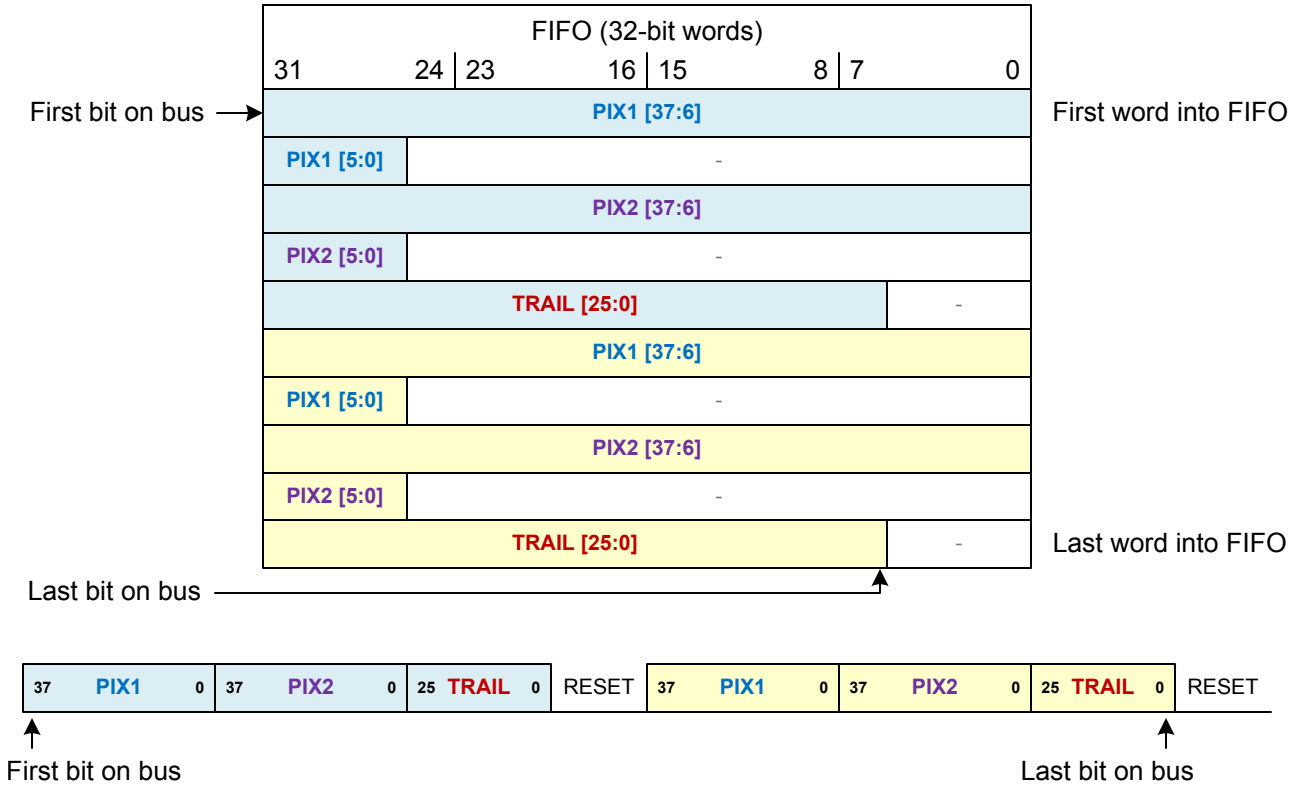


Figure 28.11. MSB-first with 32-bit Memory Alignment

28.3.9 Setup

The general order for configuring the PIXELRZ interface is as follows:

1. Configure the PIXELRZ clock source in the CMU interface.
2. Configure options for data formatting, operational mode, clock scaling, and other details in the CFG register.
3. Define the frame parameters in the FRAMECFG register.
4. Define symbol timing using the T0CFG, T1CFG, TSTOPCFG (parallel mode), and TRSTCFG (sequential mode) registers.
5. Route the output to the desired I/O using the DBUS controls in the GPIO interface.
6. Enable the module by writing EN.EN to '1'.

Once enabled, the PIXELRZ is ready to be used. The desired interrupt sources (IEN register) may be reconfigured on-the-fly, but the module should be disabled by clearing EN.EN to '0' if the basic configuration, frame, or timing parameters need to be modified.

28.4 PIXELRZ Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	PIXELRZ_IPVERSION	R	IP VERSION Register
0x004	PIXELRZ_EN	RW ENABLE	Enable Register
0x008	PIXELRZ_SWRST	RW SWRST	Software Reset Register
0x00C	PIXELRZ_CFG	RW CONFIG	Configuration Register
0x010	PIXELRZ_FRAMECFG	RW CONFIG	Frame Configuration Register
0x014	PIXELRZ_T0CFG	RW CONFIG	Zero Symbol Configuration Register
0x018	PIXELRZ_T1CFG	RW CONFIG	One Symbol Configuration Register
0x01C	PIXELRZ_TSTOPCFG	RW CONFIG	Stop Sympbol Configuration Register
0x020	PIXELRZ_TRSTCFG	RW CONFIG	Reset Symbol Configuration Register
0x024	PIXELRZ_CMD	W SYNC	Command Register
0x028	PIXELRZ_TXDATA	W	Transmit Buffer Data Register
0x02C	PIXELRZ_STATUS	RH	Status Register
0x030	PIXELRZ_IF	RWH INTFLAG	Interrupt Flag Register
0x034	PIXELRZ_IEN	RW	Interrupt Enable Register
0x1000	PIXELRZ_IPVERSION_SET	R	IP VERSION Register
0x1004	PIXELRZ_EN_SET	RW ENABLE	Enable Register
0x1008	PIXELRZ_SWRST_SET	RW SWRST	Software Reset Register
0x100C	PIXELRZ_CFG_SET	RW CONFIG	Configuration Register
0x1010	PIXELRZ_FRAMECFG_SET	RW CONFIG	Frame Configuration Register
0x1014	PIXELRZ_T0CFG_SET	RW CONFIG	Zero Symbol Configuration Register
0x1018	PIXELRZ_T1CFG_SET	RW CONFIG	One Symbol Configuration Register
0x101C	PIXELRZ_TSTOPCFG_SET	RW CONFIG	Stop Sympbol Configuration Register
0x1020	PIXELRZ_TRSTCFG_SET	RW CONFIG	Reset Symbol Configuration Register
0x1024	PIXELRZ_CMD_SET	W SYNC	Command Register
0x1028	PIXELRZ_TXDATA_SET	W	Transmit Buffer Data Register
0x102C	PIXELRZ_STATUS_SET	RH	Status Register
0x1030	PIXELRZ_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1034	PIXELRZ_IEN_SET	RW	Interrupt Enable Register
0x2000	PIXELRZ_IPVERSION_CLR	R	IP VERSION Register
0x2004	PIXELRZ_EN_CLR	RW ENABLE	Enable Register
0x2008	PIXELRZ_SWRST_CLR	RW SWRST	Software Reset Register
0x200C	PIXELRZ_CFG_CLR	RW CONFIG	Configuration Register
0x2010	PIXELRZ_FRAMECFG_CLR	RW CONFIG	Frame Configuration Register
0x2014	PIXELRZ_T0CFG_CLR	RW CONFIG	Zero Symbol Configuration Register
0x2018	PIXELRZ_T1CFG_CLR	RW CONFIG	One Symbol Configuration Register
0x201C	PIXELRZ_TSTOPCFG_CLR	RW CONFIG	Stop Sympbol Configuration Register
0x2020	PIXELRZ_TRSTCFG_CLR	RW CONFIG	Reset Symbol Configuration Register
0x2024	PIXELRZ_CMD_CLR	W SYNC	Command Register
0x2028	PIXELRZ_TXDATA_CLR	W	Transmit Buffer Data Register

Offset	Name	Type	Description
0x202C	PIXELRZ_STATUS_CLR	RH	Status Register
0x2030	PIXELRZ_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2034	PIXELRZ_IEN_CLR	RW	Interrupt Enable Register
0x3000	PIXELRZ_IPVERSION_TGL	R	IP VERSION Register
0x3004	PIXELRZ_EN_TGL	RW ENABLE	Enable Register
0x3008	PIXELRZ_SWRST_TGL	RW SWRST	Software Reset Register
0x300C	PIXELRZ_CFG_TGL	RW CONFIG	Configuration Register
0x3010	PIXELRZ_FRAMECFG_TGL	RW CONFIG	Frame Configuration Register
0x3014	PIXELRZ_T0CFG_TGL	RW CONFIG	Zero Symbol Configuration Register
0x3018	PIXELRZ_T1CFG_TGL	RW CONFIG	One Symbol Configuration Register
0x301C	PIXELRZ_TSTOPCFG_TGL	RW CONFIG	Stop Sympbol Configuration Register
0x3020	PIXELRZ_TRSTCFG_TGL	RW CONFIG	Reset Symbol Configuration Register
0x3024	PIXELRZ_CMD_TGL	W SYNC	Command Register
0x3028	PIXELRZ_TXDATA_TGL	W	Transmit Buffer Data Register
0x302C	PIXELRZ_STATUS_TGL	RH	Status Register
0x3030	PIXELRZ_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3034	PIXELRZ_IEN_TGL	RW	Interrupt Enable Register

28.5 PIXELRZ Register Description

28.5.1 PIXELRZ_IPVERSION - IP VERSION Register

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP version ID**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

28.5.2 PIXELRZ_EN - Enable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablment busy status** When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFO. Enabling the module also sets the programmed GPIO pad to its default IDLE state.
0	EN	0x0	RW	**module enable** The EN bit enables the module. When EN is cleared(disablement), it halts module operation immediately, and initializes the core domain such that when the module is re-enabled it starts cleanly.

Value	Mode	Description
0	DISABLE	Disable Peripheral Clock
1	ENABLE	Enable Peripheral Clock

28.5.3 PIXELRZ_SWRST - Software Reset Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Reset busy status** When SWRST command is issued, resetting logic sets RESETTING status immediately, and later it is cleared when reset process finishes.
0	SWRST	0x0	W	**Software Reset bit** A software reset command field resets the module back to the initial condition, similar to a power on reset.

28.5.4 PIXELRZ_CFG - Configuration Register

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset										0x0	0x0	0x0		0x0	0x0	0x0	0x0										0x0				0x0	0x0
Access										RW	RW	RW		RW	RW	RW	RW										RW				RW	RW
Name										RESETEN	STOPEN	TRIGSEL		MODE	MSBF	MEMALIGN	DIV										TXFWL				JITTEREN	OUTINV

Bit	Name	Reset	Access	Description
31:23	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
22	RESETEN	0x0	RW	**RESET Symbol Enable**
	Enables RESET symbols on the bus for sequential mode operation			
	Value	Mode		Description
	0	DISABLE		RESET symbols will not be sent
	1	ENABLE		RESET symbols will be sent at the end of each frame
21	STOPEN	0x0	RW	**STOP Symbol Enable**
	Enables STOP symbols on the bus for parallel mode operation			
	Value	Mode		Description
	0	DISABLE		STOP symbols will not be sent
	1	ENABLE		STOP symbols will be sent at the end of each pixel
20:19	TRIGSEL	0x0	RW	**Trigger Select**
	Configures the frame trigger source			
	Value	Mode		Description
	0	SW		Software trigger mode
	1	AUTO		Auto trigger mode
	2	PRS		PRS trigger mode
18	MODE	0x0	RW	**Bus Mode**
	Selects sequential or parallel bus mode			
	Value	Mode		Description
	0	SEQUENTIAL		PIXELRZ formats data for sequential LED connection
	1	PARALLEL		PIXELRZ formats data for parallel LED connection
17	MSBF	0x0	RW	**MSB First Enable**
	Defines if data is sent from FIFO LSB-first or MSB-first			
	Value	Mode		Description
	0	LSB		Data will be sent LSB-first
	1	MSB		Data will be sent MSB-first

Bit	Name	Reset	Access	Description
16	MEMALIGN	0x0	RW	**Memory Alignment**
	Defines how FIFO data is aligned			
	Value	Mode		Description
	0	ALIGN8		Use 8-bit alignment in FIFO
	1	ALIGN32		Use 32-bit alignment in FIFO
15:6	DIV	0x0	RW	**Clock Prescaler**
	Divider for the selected PIXELRZCLK. CLK_PIX = PIXELRZCLK / (DIV+1)			
5:2	TXFWL	0x0	RW	**TXDATA FIFO Watermark Level**
	Sets the watermark level for the TXDATA FIFO. If TXDATA FIFO has more free entries than this programmed level, an interrupt or DMA request will be asserted			
1	JITTEREN	0x0	RW	**Jitter Enable**
	Enable jitter on T0LOW and T1LOW time for parallel mode operation			
	Value	Mode		Description
	0	DISABLE		Jitter generation is disabled
	1	ENABLE		Jitter generation is enabled
0	OUTINV	0x0	RW	**Invert Output**
	Setting this bit will Invert the output from the block			
	Value	Mode		Description
	0	NORMAL		Output will be logic 0 during low times and logic 1 during high times
	1	INVERT		Output will be logic 1 during low times and logic 0 during high times

28.5.5 PIXELRZ_FRAMECFG - Frame Configuration Register

Offset	Bit Position																															
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset				0x0									0x0										0x0									
Access				RW									RW										RW									
Name				TRAILWIDTH									PIXELWIDTH										PIXELNUM									

Bit	Name	Reset	Access	Description
31:29	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
28:20	TRAILWIDTH	0x0	RW	**Trail Data Width**
	Sets the number of trail data bits to be sent in each frame. When cleared to 0, no trail data will be sent.			
19:10	PIXELWIDTH	0x0	RW	**Pixel Data Width**
	Defines the number of bits in each pixel. If cleared to 0, data will be sent from the FIFO continuously, until an underflow error occurs.			
9:0	PIXELNUM	0x0	RW	**Pixels per Frame**
	Defines the number of pixels per frame. If cleared to 0, blocks of data of PIXELWIDTH length will be sent from the FIFO continuously, until and underflow error occurs.			

28.5.6 PIXELRZ_T0CFG - Zero Symbol Configuration Register

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0											0x0										
Access											RW											RW										
Name											HIGH											LOW										

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:11	HIGH	0x0	RW	**Zero symbol High Time**
	Number of CLK_PIX clock cycles in DATA0 symbol high period. Minimum is 1 cycle (0 will default to 1).			
10:0	LOW	0x0	RW	**Zero symbol Low Time**
	Number of CLK_PIX clock cycles in DATA0 symbol low period. Minimum is 1 cycle (0 will default to 1).			

28.5.7 PIXELRZ_T1CFG - One Symbol Configuration Register

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0											0x0										
Access											RW											RW										
Name											HIGH											LOW										

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:11	HIGH	0x0	RW	**One Symbol High Time**
	Number of CLK_PIX clock cycles in DATA1 symbol high period. Minimum is 1 cycle (0 will default to 1).			
10:0	LOW	0x0	RW	**One Symbol Low Time**
	Number of CLK_PIX clock cycles in DATA1 symbol low period. Minimum is 1 cycle (0 will default to 1).			

28.5.8 PIXELRZ_TSTOPCFG - Stop Sympbol Configuration Register

Offset	Bit Position																															
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0											0x0										
Access											RW											RW										
Name											HIGH											LOW										

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21:11	HIGH	0x0	RW	**Stop Symbol High Time**
	Number of CLK_PIX clock cycles in STOP symbol high period.			
10:0	LOW	0x0	RW	**Stop Symbol Low Time**
	Number of CLK_PIX clock cycles in STOP symbol low period.			

28.5.9 PIXELRZ_TRSTCFG - Reset Symbol Configuration Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																0x0															
Access	RW																RW															
Name	HIGH																LOW															

Bit	Name	Reset	Access	Description
31:16	HIGH	0x0	RW	**Reset Symbol High Time**
	Number of CLK_PIX clock cycles in RESET symbol high period.			
15:0	LOW	0x0	RW	**Reset Symbol Low Time**
	Number of CLK_PIX clock cycles in RESET symbol low period.			

28.5.10 PIXELRZ_CMD - Command Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																														0x0	0x0	0x0
Access																														W(nB)	W(nB)	W(nB)
Name																														CLEARTX	TXDIS	TXEN

Bit	Name	Reset	Access	Description
31:3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	CLEARTX	0x0	W(nB)	**Clear TX fifo**
	Stops transmission immediately, resets the FSM, and flushes the fifo			
1	TXDIS	0x0	W(nB)	**Send stop condition**
	Stops transmission immediately, resets the FSM and a new data is popped from FIFO.			
0	TXEN	0x0	W(nB)	**Send start condition**
	Start transmitting data bits as they get loaded into FIFO			

28.5.11 PIXELRZ_TXDATA - Transmit Buffer Data Register

Offset	Bit Position																															
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	W																															
Name	TXDATA																															

Bit	Name	Reset	Access	Description
31:0	TXDATA	0x0	W	**TX Data FIFO**
	Transmit data FIFO input.			

28.5.12 PIXELRZ_STATUS - Status Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset											0x0	0x0										0x0	0x0	0x0				0x0	0x0	0x0	0x0	0x0
Access											R	R										R	R	R				R	R	R	R	R
Name											FRAMEDONE	PIXELCNT										CLEARTXBUSY	SYNCBUSY	TXFCNT				TXIDLE	TXFULL	TXRUNNING	TXFL	TXC

Bit	Name	Reset	Access	Description
31:22	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
21	FRAMEDONE	0x0	R	**Frame Done**
	Sets when Frames is transmitted. Deasserts when new frame starts or FIFO empties out.			
20:11	PIXELCNT	0x0	R	**Pixel Count**
	Indicates the number of pixels that have been sent on the bus during the current frame			
10	CLEARTXBUSY	0x0	R	**Clear TX Command Busy**
	Asserts upon CLEARTX cmd. Deasserts when process finishes			
9	SYNCBUSY	0x0	R	**Sync Busy**
	Synchronization ongoing			
8:5	TXFCNT	0x0	R	**FIFO Count**
	Indicates the number of valid entries in FIFO			
4	TXIDLE	0x0	R	**Transmitter Idle**
	Peripheral is idle and can start new transmission.			
3	TXFULL	0x0	R	**TX FIFO full**
	Transmit FIFO is full			
2	TXRUNNING	0x0	R	**TX in Progress**
	Transmssion of data is ongoing.			
1	TXFL	0x0	R	**TX FIFO Watermark**
	Indicates the level of the transmit FIFO with respect to the TXFL watermark level. Set to 1 when the empty FIFO slots are greater than TXFL.			
0	TXC	0x0	R	**TX Complete**
	Set when last "word" written to the FIFO has been transmitted out . Cleared when a new transmission starts.			

28.5.13 PIXELRZ_IF - Interrupt Flag Register

Offset	Bit Position																															
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													TXFUF	TXFOF	TXFL	TXC

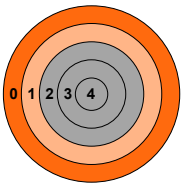
Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	TXFUF	0x0	RW	**Underflow**
	Set when trying to read from the TXFIFO when it is empty.			
2	TXFOF	0x0	RW	**Overflow**
	Set when a write is done to the TX FIFO while it is full. The data already in the TX FIFO is preserved.			
1	TXFL	0x0	RW	**Reached watermark**
	Sets when FIFO crosses watermark level			
0	TXC	0x0	RW	**Transmission complete**
	Set when Frame is compted and no more entries availble in FIFO (AUTO,PRS). Gets set after every frame in SW trigger mode			

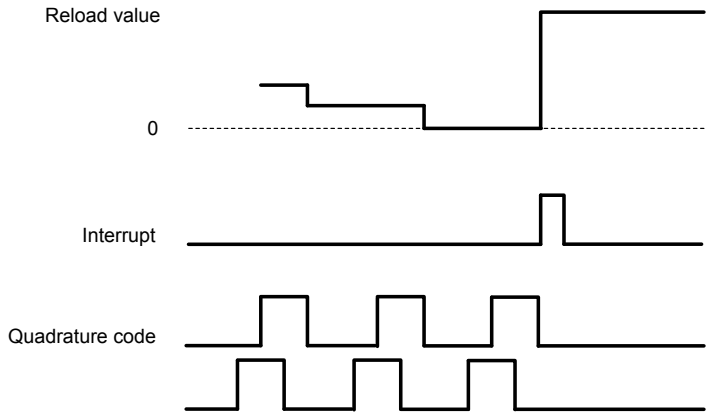
28.5.14 PIXELRZ_IEN - Interrupt Enable Register

Offset	Bit Position																															
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																													0x0	0x0	0x0	0x0
Access																													RW	RW	RW	RW
Name																													TXFUF	TXFOF	TXFL	TXC

Bit	Name	Reset	Access	Description
31:4	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
3	TXFUF	0x0	RW	**Underflow interrupt enable**
	TXDATA FIFO Underflow interrupt enable			
2	TXFOF	0x0	RW	**Overflow interrupt enable**
	TXDATA FIFO Overflow interrupt enable			
1	TXFL	0x0	RW	**Reached watermark interrupt enable**
	TXDATA FIFO watermark crossing interrupt enable			
0	TXC	0x0	RW	**Transmission complete interrupt enable**
	Transmit complete interrupt enable			

29. PCNT - Pulse Counter





Quick Facts

What?

The PCNT decodes incoming pulses. The module has a quadrature mode which may be used to decode the speed and direction of a mechanical shaft. PCNT can operate in EM0 and EM1.

Why?

The PCNT generates an interrupt after a specific number of pulses (or rotations), eliminating the need for timing or I/O interrupts and CPU processing to measure pulse widths, etc.

How?

PCNT uses the EM23GRPACLK or may be externally clocked from a pin. The module incorporates a 16-bit up/down-counter to keep track of incoming pulses or rotations.

29.1 Introduction

The Pulse Counter (PCNT) can be used for counting incoming pulses on a single input or to decode quadrature encoded inputs in EM0 and EM1. It can run from the internal EM23GRPACLK clock source while counting pulses on the PCNTn.S0IN pin. Alternatively, the PCNTn.S0IN pin or a PRS signal may be used as an external clock source that runs the PCNT counter and register access.

29.2 Features

- 16-bit counter with reload register
- Auxiliary counter for counting a single direction
- Single input oversampling up/down counter mode
- Externally clocked single input pulse up/down counter mode
- Quadrature decoder modes
  - Externally clocked quadrature decoder 1X mode
  - Oversampling quadrature decoder 1X, 2X and 4X modes
- Interrupt on counter underflow and overflow
- Interrupt when a direction change is detected (quadrature decoder mode only)
- Optional pulse width filter
- Optional input inversion/edge detect select
- Optional inputs from PRS

29.3 Functional Description

An overview of the PCNT module is shown in Figure 29.1. PCNT Overview.

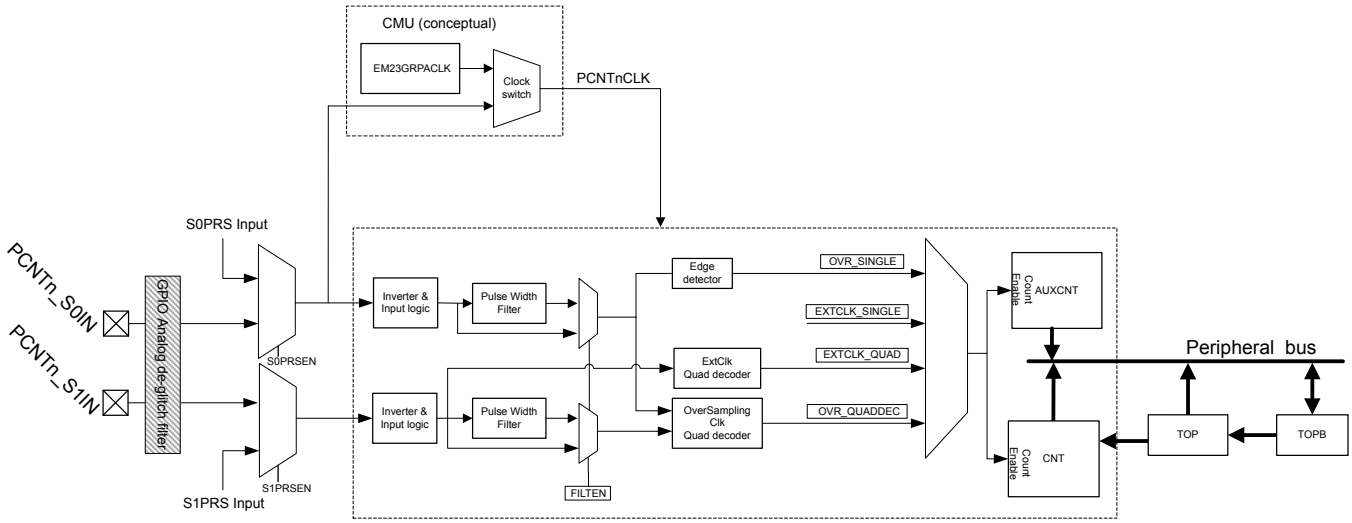


Figure 29.1. PCNT Overview

29.3.1 Pulse Counter Modes

The pulse counter can operate in single input oversampling mode (OVSSINGLE), externally clocked single input counter mode (EXTCLKSINGLE), externally clocked quadrature decoder mode (EXTCLKQUAD) and oversampling quadrature decoder modes(OVSQUAD1X, OVSQUAD2X and OVSQUAD4X). The following sections describe operation of each of these modes and how they are enabled.

29.3.1.1 Single Input Oversampling Mode

This mode is selected by writing OVSSINGLE to the MODE field in the PCNTn_CFG register. The STARTCNT bit in PCNTn_CMD is used to start the counter, and STOPCNT can be used to stop the counter. The EM23GRPACLK clock source to the pulse counter is selected by setting CLKSEL in the CMU_PCNT0CLKCTRL register to EM23GRPACLK. In this mode the maximum input toggle frequency should be 2 times slower than the frequency of the selected EM23GRPACLK clock source.

The optional pulse width filter is enabled by setting the FILTEN bit in the PCNTn_CFG register. Additionally, the PCNTn.S0IN input may be inverted, so that falling edges are counted, by setting the EDGE bit in the PCNTn_CTRL register.

If S1CDIR in the PCNTn_CTRL register is cleared, PCNTn.S0IN is the only observed input in this mode. The PCNTn.S0IN input is sampled by the PCNTnCLK and the number of detected positive or negative edges on PCNTn.S0IN appears in PCNTn_CNT. By default the counter will count up, but the counter may be configured to count down by setting the CNTDIR bit in PCNTn_CTRL.

The counting direction can also be controlled externally in this mode, by setting S1CDIR. This will make the input value on PCNTn.S1IN decide the direction counted for each PCNTn.S0IN edge. When PCNTn.S1IN is high, the count is done according to CNTDIR in PCNTn_CTRL. When PCNTn.S1IN is low, the count direction is opposite.

29.3.1.2 Externally Clocked Single Input Counter Mode

This mode is enabled by writing EXTCLKSINGLE to the MODE field in the PCNTn_CFG register. The STARTCNT bit in PCNTn_CMD is used to start the counter, and STOPCNT can be used to stop the counter. The external pin clock source is selected by setting CLKSEL in the CMU_PCNT0CLKCTRL register to PCNTS0.

Positive edges on PCNTn.S0IN are used to clock the counter. Similar to the oversampled mode, PCNTn.S1IN is used to determine the count direction if S1CDIR is set. If not, CNTDIR in PCNTn_CTRL solely defines count direction.

The digital pulse width filter is not available in this mode. The analog de-glitch filter in the GPIO pads is capable of removing some unwanted noise. However, this mode may be susceptible to spikes and unintended pulses from devices such as mechanical switches, and is therefore most suited to take input from electronic sensors etc. that generate single wire pulses.

29.3.1.3 Quadrature Decoder Modes

Two different types of quadrature decoding are supported in the pulse counter: the externally clocked (Asynchronous) quadrature decoding and the oversampling (Synchronous) quadrature decoding. The externally clocked mode supports 1X quadrature decoding whereas the oversampling mode supports 1X, 2X and 4X quadrature decoding. These modes are described in detail in 29.3.1.4 Externally Clocked Quadrature Decoder Mode and 29.3.1.5 Oversampling Quadrature Decoder Mode .

29.3.1.4 Externally Clocked Quadrature Decoder Mode

This mode is enabled by writing EXTCLKQUAD to the MODE field in PCNTn_CFG. The STARTCNT bit in PCNTn_CMD is used to start the counter, and STOPCNT can be used to stop the counter. The external pin clock source is selected by setting CLKSEL in the CMU_PCNT0CLKCTRL register to PCNTS0.

In this mode, both edges on PCNTn.S0IN pin are used to sample the PCNTn.S1IN pin, in order to decode the quadrature code. A quadrature coded signal contains information about the relative speed and direction of a rotating shaft as illustrated by Figure 29.2. PCNT Quadrature Coding, hence the direction of the counter register PCNTn_CNT is controlled automatically.

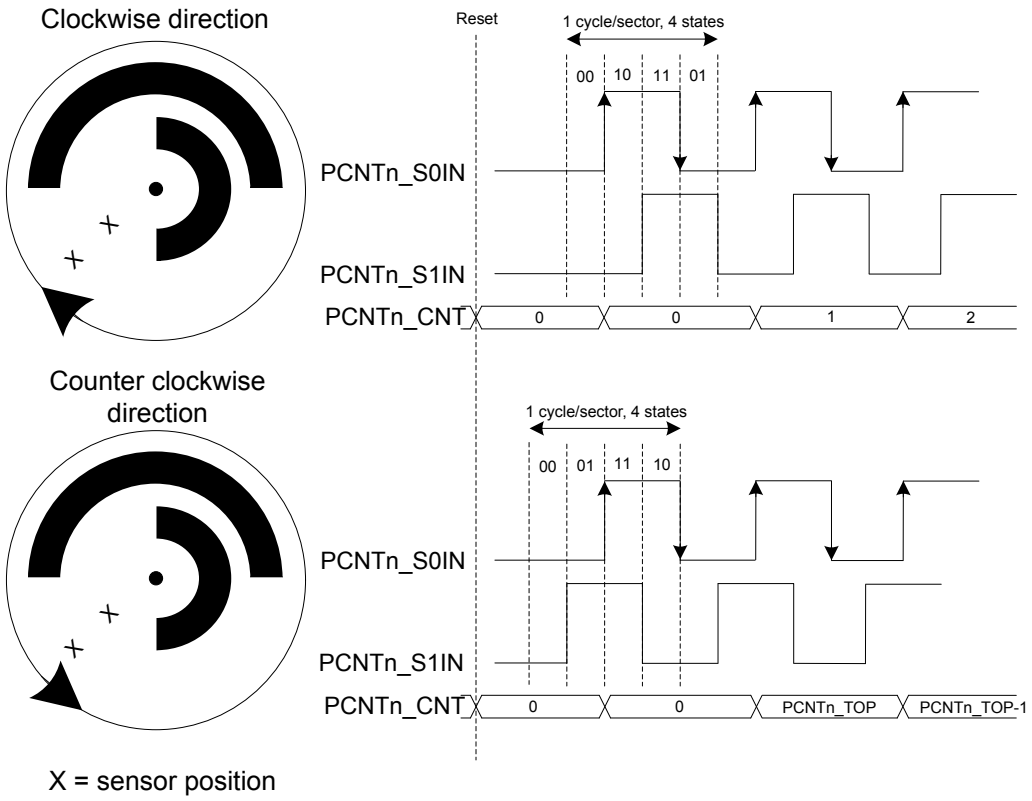


Figure 29.2. PCNT Quadrature Coding

If PCNTn.S0IN leads PCNTn.S1IN in phase, the direction is clockwise, and if it lags in phase the direction is counter-clockwise. Default behavior is illustrated by Figure 29.2. PCNT Quadrature Coding.

The counter direction may be read from the DIR bit in the PCNTn_STATUS register. Additionally, the DIRCNG interrupt in the PCNTn_IF register is generated when a direction change is detected. When a change is detected, the DIR bit in the PCNTn_STATUS register must be read to determine the current new direction.

Note: The sector disc illustrated in the figure may be finer grained in some systems. Typically, they may generate 2-4 PCNTn.S0IN wave periods per 360° rotation.

The direction of the quadrature code and control of the counter is generated by the simple binary function outlined by Table 29.1. PCNT QUAD Mode Counter Control Function. Note that this function also filters some invalid inputs that may occur when the shaft changes direction or temporarily toggles direction.

Table 29.1. PCNT QUAD Mode Counter Control Function

Inputs		Control/Status	
S1IN posedge	S1IN negedge	Count Enable	CNTDIR status bit
0	0	0	0

Inputs		Control/Status	
S1IN posedge	**S1IN negedge**	**Count Enable**	**CNTDIR status bit**
0	1	1	0
1	0	1	1
1	1	0	0
Note: PCNTn.S1IN is sampled on both edges of PCNTn.S0IN.			

29.3.1.5 Oversampling Quadrature Decoder Mode

There are three Oversampling Quadrature Decoder Modes supported: 1X , 2X and 4X. These modes are enabled by writing OVSQUAD1X, OVSQUAD2X and OVSQUAD4X, respectively, to the MODE field in PCNTn_CFG. The STARTCNT bit in PCNTn_CMD is used to start the counter, and STOPCNT can be used to stop the counter. The EM23GRPACLK clock source to the pulse counter must be selected by setting CLKSEL in the CMU_PCNT0CLKCTRL register to EM23GRPACLK.

The optional pulse width filter is enabled by setting the FILTEN bit in the PCNTn_CFG register. The filter applies to both inputs PCNTn.S0IN and PCNTn.S1IN. The filter length is configured by FILTLEN in PCNTn_OVSCTRL register.

Based on the modes selected, the decoder updates the counter on different events. In the OVSQUAD1X mode, the counter is updated on the rising edge of the PCNTn.S0IN input when counting up, and on the negedge of the PCNTn.S0IN input when counting down. In the OVSQUAD2X mode, the counter is updated on both edges of PCNTn.S0IN input. In the OVSQUAD4X mode the counter is updated on both edges of both inputs PCNTn.S0IN and PCNTn.S1IN. Table 29.2. PCNT OVSQUAD 1X, 2X and 4X Mode Counter Control Function outlines the increment or decrement of the counter based on the Quadrature Mode selected.

Note: The decoding behavior of OVSQUAD1X mode is slightly different compared to EXTCLKQUAD mode(also 1X mode). In the EXTCLKQUAD mode, the counter is updated only on the posedge of S0IN input. However, in the OVSQUAD1X mode, the counter is updated on the posedge of S0IN when counting up and on the negedge of S0IN when counting down.

Table 29.2. PCNT OVSQUAD 1X, 2X and 4X Mode Counter Control Function

Direction	Previous State		Next State		OVSQUAD MODE		
	S1IN	S0IN	S1IN	S0IN	1X	2X	4X
Clockwise	0	0	0	1	+1	+1	+1
	0	1	1	1			+1
	1	1	1	0		+1	+1
	1	0	0	0			+1
Counter Clockwise	1	0	1	1		-1	-1
	1	1	0	1			-1
	0	1	0	0	-1	-1	-1
	0	0	1	0			-1

Figure 29.3. PCNT State Transitions for Different Oversampling Quadrature Decoder Modes illustrates the different states of the quadrature input and the state transitions that updates the counter for the different modes. Each cycle of the input states results in 1 update, 2 updates and 4 updates of the counter for OVSQUAD1X, OVSQUAD2X and OVSQUAD4X modes respectively.

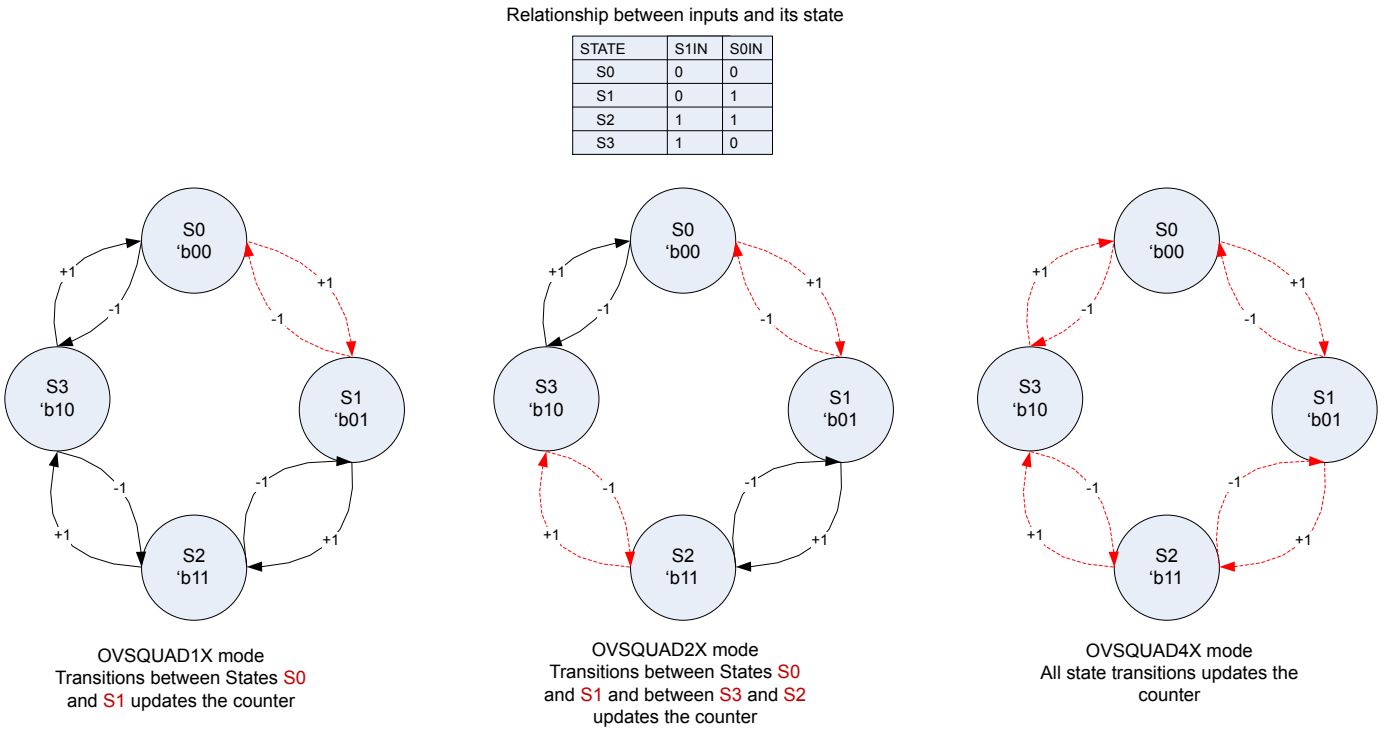


Figure 29.3. PCNT State Transitions for Different Oversampling Quadrature Decoder Modes

The counter direction can be read from the DIR bit in PCNTn_STATUS register. Additionally, the DIRCNG interrupt in the PCNTn_IF is generated when the direction change is detected. When a change is detected, the DIR bit in the PCNTn_STATUS register must be read to determine the new direction.

In the oversampling quadrature decoder modes, the maximum input toggle frequency supported is PCNTnCLK / 4. For frequencies above PCNTnCLK / 4, incorrect decoding occurs. The different decoding modes and the counter updates are further illustrated by Figure 29.4. PCNT Oversampling Quadrature Decoder 1X Mode, Figure 29.5. PCNT Oversampling Quadrature Decoder 2X Mode and Figure 29.6. PCNT Oversampling Quadrature Decoder 4X Mode.

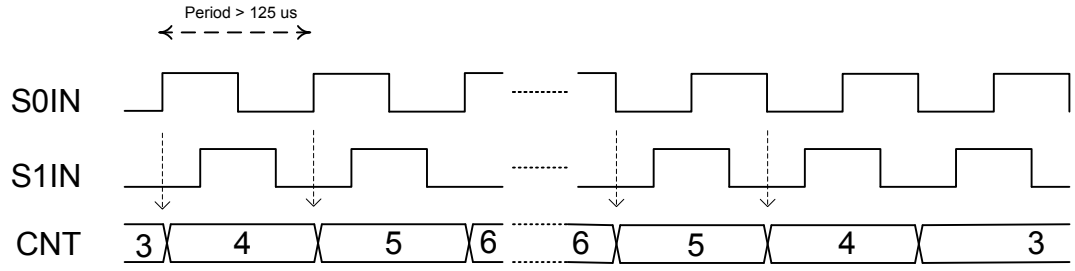


Figure 29.4. PCNT Oversampling Quadrature Decoder 1X Mode

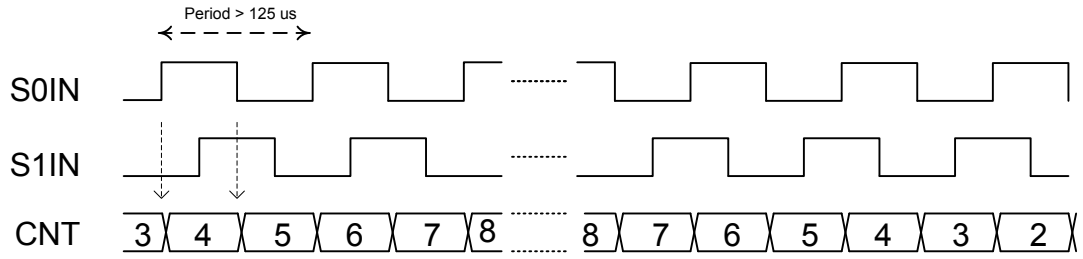


Figure 29.5. PCNT Oversampling Quadrature Decoder 2X Mode

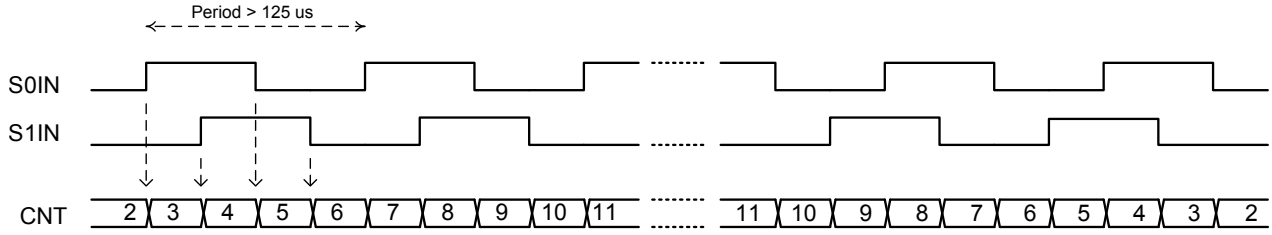


Figure 29.6. PCNT Oversampling Quadrature Decoder 4X Mode

The above modes, by default, are prone to flutter effects in the inputs PCNTn.S0IN and PCNTn.S1IN. When this occurs, the counter changes directions rapidly causing DIRCNG interrupts and unnecessarily waking the core. To prevent this, set FLUTTERRM in the PCNTn_OVSCTRL register. When enabled, flutter is removed, thus preventing unnecessary wakeup of the core. The flutter removal logic works by preventing update of the counter value if the wheel keeps changing direction as a result of flutter. The counter is only updated if the current and previous state transition of the rotation are in the same direction. These state transitions are quadrature decoder mode specific. The highlighted state transitions in Figure 29.3. PCNT State Transitions for Different Oversampling Quadrature Decoder Modes are the ones considered for the different quadrature decoder modes. Figure 29.7. PCNT Oversampling Quadrature Decoder with Flutter Removal shows how the counter is updated for the different quadrature decoder modes with flutter removal FLUTTERRM enabled in PCNTn_OVSCTRL.

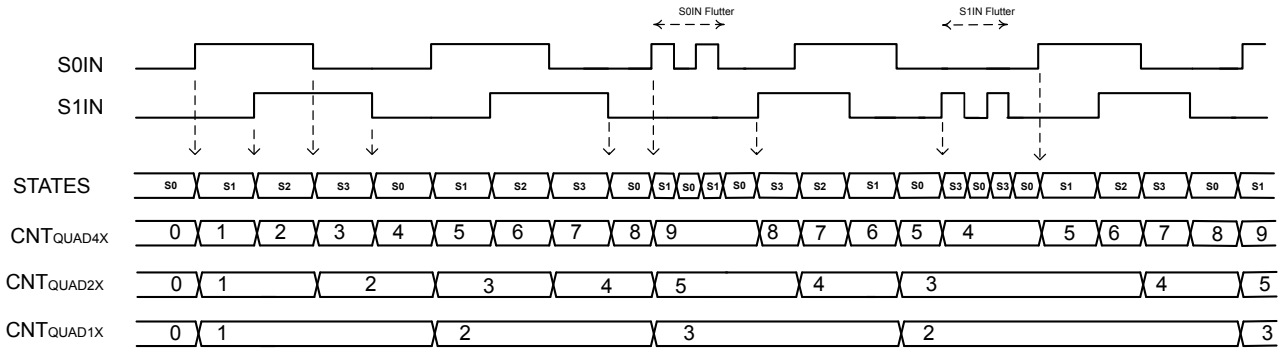


Figure 29.7. PCNT Oversampling Quadrature Decoder with Flutter Removal

29.3.2 Hysteresis

By default the pulse counter wraps to 0 when passing the configured top value, and wraps to the top value when counting down from 0. On these events, a system will likely want to wake up to store and track the overflow count. This is fine if the pulse counter is tracking a monotonic value or a value that does not change directions frequently. In the latter scenario, if the counter changes directions around the overflow/underflow point, the system will have to wake up frequently to keep track of the rotations, resulting in higher current consumption.

To solve this, the pulse counter has a way of introducing hysteresis to the counter. When HYST in PCNTn_CFG is set, the pulse counter will always wrap to TOP/2 on underflows and overflows. This takes the counter away from the area where it might overflow or underflow, removing the problem. Figure 29.8. PCNT Hysteresis Behavior of Counter illustrates the hysteresis behavior.

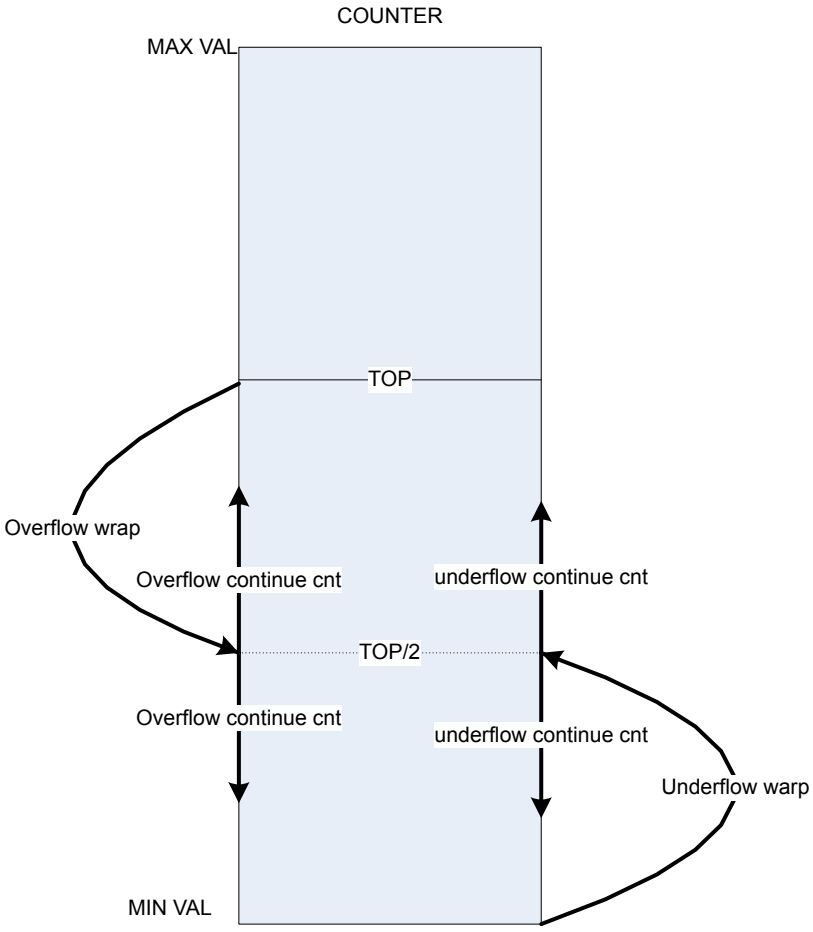


Figure 29.8. PCNT Hysteresis Behavior of Counter

Given a starting value of 0 for the counter, the absolute count value when hysteresis is enabled can be calculated with the equations Figure 29.9. Absolute Position With Hysteresis and Even TOP Value or Figure 29.10. Absolute Position With Hysteresis and Odd TOP Value, depending on whether the TOP value is even or odd.

$$CNT_{abs} = CNT - UF_{CNT} \times (TOP/2+1) + OF_{CNT} \times (TOP/2+1)$$

Figure 29.9. Absolute Position With Hysteresis and Even TOP Value

$$CNT_{abs} = CNT - UF_{CNT} \times (TOP/2+1) + OF_{CNT} \times (TOP/2+2)$$

Figure 29.10. Absolute Position With Hysteresis and Odd TOP Value

29.3.3 Auxiliary Counter

To be able to keep explicit track of counting in one direction in addition to the regular counter which counts both up and down, the auxiliary counter can be used. The pulse counter can, for instance, be configured to keep track of the absolute rotation of the wheel, while at the same time the auxiliary counter can keep track of how much the wheel has reversed.

The auxiliary counter is enabled by configuring AUXCNTEV in PCNTn_CTRL. It will always count up, but it can be configured whether it should count up on up-events, down-events or both, keeping track of rotation either way or general movement. The value of the auxiliary counter can be read from the PCNTn_AUXCNT register.

Overflows on the auxiliary counter happen when the auxiliary counter passes the top value of the pulse counter, configured in PCNTn_TOP. In that event, the AUXOF interrupt flag is set, and the auxiliary counter wraps to 0.

The auxiliary counter is started and stopped in a similar way to the main counter, using commands in the PCNTn_CMD register. The STARTAUXCNT bit in PCNTn_CMD is used to start the auxiliary counter, and STOPAUXCNT can be used to stop the auxiliary counter.

As the auxiliary counter, the main counter can be configured to count only on certain events. This is done through CNTEV in PCNTn_CTRL, and it is possible like for the auxiliary counter, to make the main counter count on only up and down events. The difference between the counters is that where the auxiliary counter will only count up, the main counter will count up or down depending on the direction of the count event.

29.3.4 Register Access

The counter-clock domain may be clocked externally. To update the counter-clock domain registers from software in this mode, 2-3 clock pulses on the external clock are needed to synchronize accesses to the externally clocked domain. Clock source switching is controlled from the registers in the CMU.

When the CORERST bit in the PCNTn_CMD register is set by software, the PCNT clock domain is synchronously reset and released two PCNT clock edges later. This synchronous reset restores the reset values in PCNTn_TOP, PCNTn_CNT and other core registers in the PCNT clock domain.

CNTRST works in a similar manner as CORERST, but only resets the counter, CNT. Note that the counter is also reset by CORERST.

AUXCNTRST works in a similar manner as CORERST, but only resets the auxiliary counter, PCNTn_AUXCNT. Note that the auxiliary counter is also reset by CORERST.

Note: PCNTn_CNT is a read-only register. When writing to PCNTn_TOP, make sure that the counter value, PCNTn_CNT, can not exceed the value written to PCNTn_TOP within two clock cycles.

Note: To ensure reset during SWRST, if the counter-clock domain is clocked externally, it is advisable to swith the counter-clock to the internal clock before performing the software reset. The clock pulses from an external source cannot be relied upon. By switching to an internal clock the software reset operation completion will be guaranteed.

29.3.5 Clock Sources

The pulse counter may be clocked from two possible clock sources: EM23GRPACLK or an external clock. The clock selection is configured by setting the CLKSEL field in the CMU_PCNTnCLKCTRL register. The default clock source is the EM23GRPACLK.

This PCNT module may also use PCNTn.S0IN as an external clock to clock the counter (EXTCLKSINGLE mode) and to sample PCNTn.S1IN (EXTCLKQUAD mode). Setup, hold and max frequency constraints for PCNTn.S0IN and PCNTn.S1IN for these modes are specified in the device data sheet.

Note: If changing PCNT to an external clock source, either through the PCNTn.S0IN or S0PRS input, the CMU clock switch will need at least 2 external clocks to sync over to that new clock source. Register will not load and counting will not begin until there are few external clocks.

Note: Switching external clock source to the S0PRS input can cause a clock glitch. So S0PRSEN in PCNTn_CFG should be configured before configuring the CLKSEL field in CMU_PCNTnCLKCTRL.

29.3.6 Input Filter

An optional pulse width filter is available in OVSSINGLE and OVSQUAD modes, when EM23GRPACLK is selected as a clock source for the Pulse Counter. The filter is enabled by writing 1 to the FILTEN bit in the PCNTn_CTRL register. When enabled, the high and low periods of PCNTn.S0IN and PCNTn.S1IN must be stable for a programmable number of consecutive clock cycles before the edge is passed to the edge detector. The filter length should be programmed in FILTLEN field of the PCNTn_OVSCTRL register.

The filter length is given by Figure 29.11. PCNT Input Filter Length Equation:

Filter length = (FILTLEN + 5) EM23GRPACLK cycles

Figure 29.11. PCNT Input Filter Length Equation

The maximum filter length configured is 260 EM23GRPACLK cycles.

In EXTCLKSINGLE and EXTCLKQUAD mode, there is no digital pulse width filter available.

29.3.7 Edge Polarity

The edge polarity can be set by configuring the EDGE bit in the PCNTn_CTRL register. When this bit is cleared, the pulse counter counts positive edges of PCNTn.S0IN input. When this bit is set, the pulse counter counts negative edges in OVSSINGLE mode. Also, when the EDGE bit is set in the OVSSINGLE and EXTCLKSINGLE modes, the PCNTn.S1IN input is inverted. In OVSQUAD 1X-4X modes the EDGE bit inverts both inputs.

Note: The EDGE bit in PCNTn_CTRL has no effect in EXTCLKQUAD mode.

29.3.8 PRS and PCNTn.S0IN,PCNTn.S1IN Inputs

It is possible to receive input from a PRS signal on both PCNTn.S0IN (or PCNTn.S1IN) by setting S0PRSEN (or S1PRSEN) in PCNTn_CFG. Routing of PRS channels to the inputs is performed using the PRS_CONSUMER_PCNTn_S0IN and PRS_CONSUMER_PCNTn_S1IN registers.

In the Oversampling quadrature decoder modes, the input frequency should be less than 4 times slower than the sampling clock PCNTnCLK to ensure correct functionality.

In the Single Input Oversampling Mode, the input toggle frequency should be 2 times slower than the sampling clock PCNTnCLK to ensure correct functionality.

In the externally clocked modes (where S0IN or S0PRS is used as PCNTnCLK clock source), the input frequency should be less than 1 MHz to ensure correct functionality.

The PCNT module generates two PRS signals, the PCNTn UFOF signal and the PCNTn DIR signal. The PCNTn UFOF signal is generated when the counter overflows or underflows. The PCNTn DIR signal is a level, and indicates the current direction of count of counter CNT.

29.3.9 Interrupts

The interrupt generated by PCNT uses the PCNTn interrupt vector. Software must read the PCNTn_IF register to determine which module interrupt that generated the vector invocation.

29.3.9.1 Underflow and Overflow Interrupts

The underflow interrupt flag (UF) is set when the counter counts down from 0 (i.e. when the value of the counter is 0 and a new pulse is received). The PCNTn_CNT register is loaded with the PCNTn_TOP value after this event.

The overflow interrupt flag (OF) is set when the counter counts up from the PCNTn_TOP (reload) value (i.e. if PCNTn_CNT = PCNTn_TOP and a new pulse is received). The PCNTn_CNT register is loaded with the value 0 after this event.

29.3.9.2 Direction Change Interrupt

The PCNTn module sets the DIRCNG interrupt flag (PCNTn_IF register) for EXTCLKQUAD and OVSQUAD1X-4X modes when the direction of the quadrature code changes. The behavior of this interrupt in the EXTCLKQUAD mode is illustrated by Figure 29.12. PCNT Direction Change Interrupt (DIRCNG) Generation.

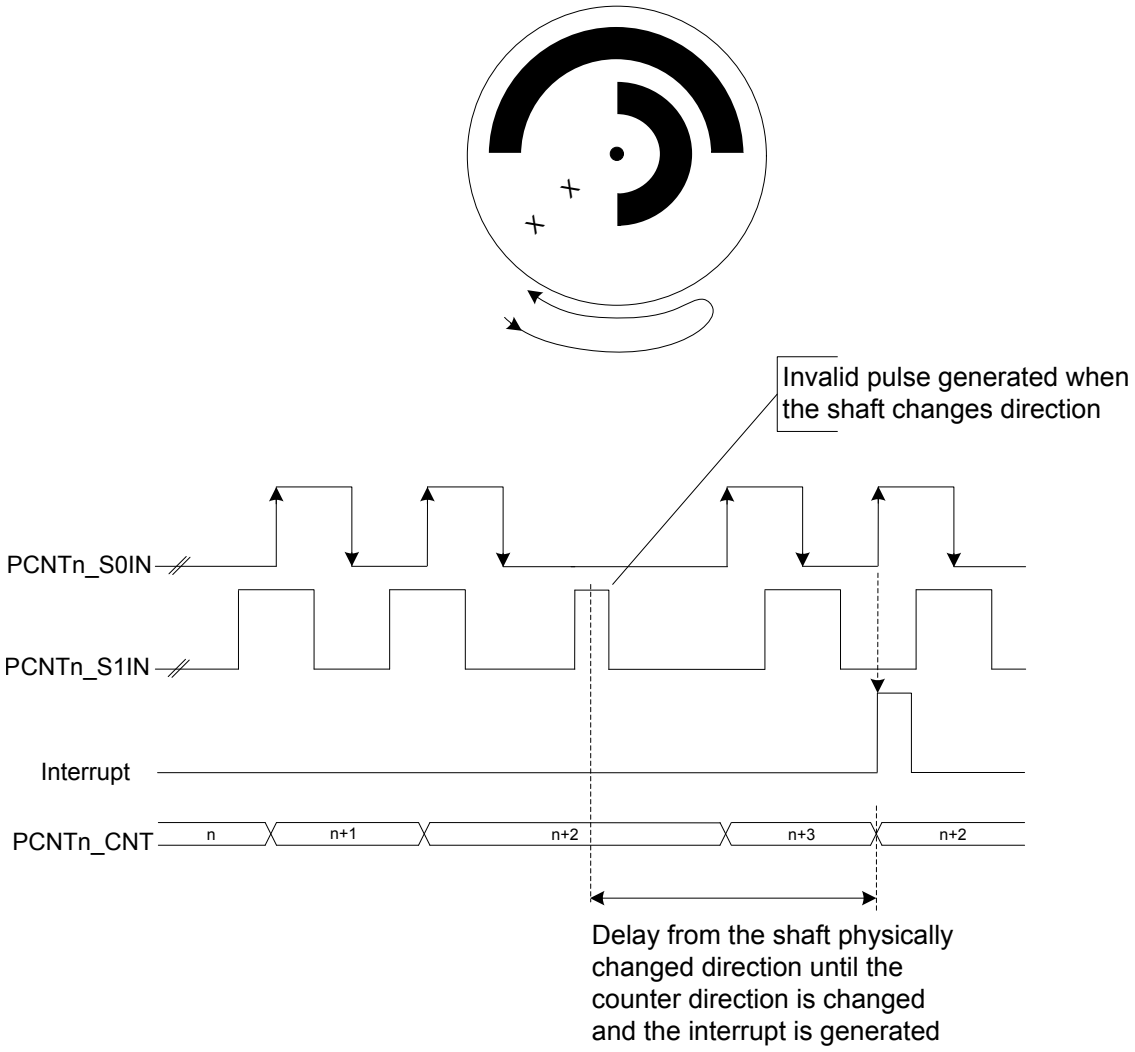


Figure 29.12. PCNT Direction Change Interrupt (DIRCNG) Generation

29.4 PCNT Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	PCNT_IPVERSION	R	IP Version ID
0x004	PCNT_EN	RW ENABLE	Module Enable Register
0x008	PCNT_SWRST	RW SWRST	Software Reset Register
0x00C	PCNT_CFG	RW CONFIG	Configuration Register
0x010	PCNT_CTRL	RW LFSYNC	Control Register
0x014	PCNT_CMD	W LFSYNC	Command Register
0x018	PCNT_STATUS	RH	Status Register
0x01C	PCNT_IF	RWH INTFLAG	Interrupt Flag Register
0x020	PCNT_IEN	RW	Interrupt Enable Register
0x024	PCNT_CNT	RH	Counter Value Register
0x028	PCNT_AUXCNT	RH	Auxiliary Counter Value Register
0x02C	PCNT_TOP	RWH LFSYNC	Top Value Register
0x030	PCNT_TOPB	RW LFSYNC	Counter Top Value Buffer Register
0x034	PCNT_OVSCTRL	RW LFSYNC	Oversampling Control Register
0x038	PCNT_SYNCBUSY	RH	Synchronization Busy Register
0x03C	PCNT_LOCK	W	Configuration Lock Register
0x1000	PCNT_IPVERSION_SET	R	IP Version ID
0x1004	PCNT_EN_SET	RW ENABLE	Module Enable Register
0x1008	PCNT_SWRST_SET	RW SWRST	Software Reset Register
0x100C	PCNT_CFG_SET	RW CONFIG	Configuration Register
0x1010	PCNT_CTRL_SET	RW LFSYNC	Control Register
0x1014	PCNT_CMD_SET	W LFSYNC	Command Register
0x1018	PCNT_STATUS_SET	RH	Status Register
0x101C	PCNT_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x1020	PCNT_IEN_SET	RW	Interrupt Enable Register
0x1024	PCNT_CNT_SET	RH	Counter Value Register
0x1028	PCNT_AUXCNT_SET	RH	Auxiliary Counter Value Register
0x102C	PCNT_TOP_SET	RWH LFSYNC	Top Value Register
0x1030	PCNT_TOPB_SET	RW LFSYNC	Counter Top Value Buffer Register
0x1034	PCNT_OVSCTRL_SET	RW LFSYNC	Oversampling Control Register
0x1038	PCNT_SYNCBUSY_SET	RH	Synchronization Busy Register
0x103C	PCNT_LOCK_SET	W	Configuration Lock Register
0x2000	PCNT_IPVERSION_CLR	R	IP Version ID
0x2004	PCNT_EN_CLR	RW ENABLE	Module Enable Register
0x2008	PCNT_SWRST_CLR	RW SWRST	Software Reset Register
0x200C	PCNT_CFG_CLR	RW CONFIG	Configuration Register
0x2010	PCNT_CTRL_CLR	RW LFSYNC	Control Register
0x2014	PCNT_CMD_CLR	W LFSYNC	Command Register
0x2018	PCNT_STATUS_CLR	RH	Status Register

Offset	Name	Type	Description
0x201C	PCNT_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x2020	PCNT_IEN_CLR	RW	Interrupt Enable Register
0x2024	PCNT_CNT_CLR	RH	Counter Value Register
0x2028	PCNT_AUXCNT_CLR	RH	Auxiliary Counter Value Register
0x202C	PCNT_TOP_CLR	RWH LFSYNC	Top Value Register
0x2030	PCNT_TOPB_CLR	RW LFSYNC	Counter Top Value Buffer Register
0x2034	PCNT_OVSCTRL_CLR	RW LFSYNC	Oversampling Control Register
0x2038	PCNT_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x203C	PCNT_LOCK_CLR	W	Configuration Lock Register
0x3000	PCNT_IPVERSION_TGL	R	IP Version ID
0x3004	PCNT_EN_TGL	RW ENABLE	Module Enable Register
0x3008	PCNT_SWRST_TGL	RW SWRST	Software Reset Register
0x300C	PCNT_CFG_TGL	RW CONFIG	Configuration Register
0x3010	PCNT_CTRL_TGL	RW LFSYNC	Control Register
0x3014	PCNT_CMD_TGL	W LFSYNC	Command Register
0x3018	PCNT_STATUS_TGL	RH	Status Register
0x301C	PCNT_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x3020	PCNT_IEN_TGL	RW	Interrupt Enable Register
0x3024	PCNT_CNT_TGL	RH	Counter Value Register
0x3028	PCNT_AUXCNT_TGL	RH	Auxiliary Counter Value Register
0x302C	PCNT_TOP_TGL	RWH LFSYNC	Top Value Register
0x3030	PCNT_TOPB_TGL	RW LFSYNC	Counter Top Value Buffer Register
0x3034	PCNT_OVSCTRL_TGL	RW LFSYNC	Oversampling Control Register
0x3038	PCNT_SYNCBUSY_TGL	RH	Synchronization Busy Register
0x303C	PCNT_LOCK_TGL	W	Configuration Lock Register

29.5 PCNT Register Description

29.5.1 PCNT_IPVERSION - IP Version ID

Offset	Bit Position																															
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP VERSION**
	Gives access to the IP VERSION of PCNT			

29.5.2 PCNT_EN - Module Enable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disablement busy status** When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except the INTFLAG register.
0	EN	0x0	RW	**PCNT Module Enable** Enable the PCNT module. When EN is cleared(disablement), it halts module operation immediately, and initialize the core domain such that when the is re-enabled, it starts cleanly.

29.5.3 PCNT_SWRST - Software Reset Register

Offset	Bit Position																													
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	W
Name																															RESETTING	SWRST

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	RESETTING	0x0	R	**Software reset busy status** When SWRST command is issued, resetting logic sets RESETTING status immediately, and later it is cleared when reset process finishes.
0	SWRST	0x0	W	**Software reset command** A software reset command field resets the module back to the initial condition, similar to a power on reset condition

29.5.4 PCNT_CFG - Configuration Register

Offset	Bit Position																															
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																							0x0	0x0		0x0	0x0	0x0		0x0		
Access																							RW	RW		RW	RW	RW		RW		
Name																							S1PRSEN	S0PRSEN		HYST	FILTEN	DEBUGHALT		MODE		

Bit	Name	Reset	Access	Description
31:10	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
9	S1PRSEN	0x0	RW	**S1IN PRS Enable**
	When set, the PRS channel is selected as input to S1IN.			
8	S0PRSEN	0x0	RW	**S0IN PRS Enable**
	When set, the PRS channel is selected as input to S0IN.			
7	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
6	HYST	0x0	RW	**Enable Hysteresis**
	When hysteresis is enabled, the PCNT will always overflow and underflow to TOP/2.			
5	FILTEN	0x0	RW	**Enable Digital Pulse Width Filter**
	The filter passes all high and low periods that are at least (FILTLEN+5) clock cycles wide. This filter is only available in OVSSINGLE,OVSQUAD1X-4X modes.			
4	DEBUGHALT	0x0	RW	**Debug Mode Halt Enable**
	Set to halt the PCNT in debug mode only in OVSSINGLE and OVSQUAD modes. When in EXTCLKSINGLE or EXTCLKQUAD modes, DEBUGHALT does not halt the Pulse Counter.			

Value	Mode	Description
0	DISABLE	PCNT is running in debug mode.
1	ENABLE	PCNT is frozen in debug mode.

3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2:0	MODE	0x0	RW	**Mode Select**
	Selects the mode of operation. The corresponding clock source must be selected from the CMU.			

Value	Mode	Description
0	OVSSINGLE	Single input EM23GRPACLK oversampling mode
1	EXTCLKSINGLE	Externally clocked single input counter mode
2	EXTCLKQUAD	Externally clocked quadrature decoder mode
3	OVSQUAD1X	EM23GRPACLK oversampling quadrature decoder 1X mode
4	OVSQUAD2X	EM23GRPACLK oversampling quadrature decoder 2X mode
5	OVSQUAD4X	EM23GRPACLK oversampling quadrature decoder 4X mode

29.5.5 PCNT_CTRL - Control Register

Offset	Bit Position																													
0x010	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																									0x0		0x0			0x0	0x0	0x0
Access																									RW		RW			RW	RW	RW
Name																									AUXCNTEV		CNTEV			EDGE	CNTDIR	S1CDIR

Bit	Name	Reset	Access	Description
31:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:6	AUXCNTEV	0x0	RW	**Controls When the Aux Counter Counts**
	Selects whether the auxiliary counter responds to up-count events, down-count events or both			
	Value	Mode		Description
	0	BOTH		Counts up on both up-count and down-count events.
	1	UP		Counts up on up-count events.
	2	DOWN		Counts up on down-count events.
5:4	CNTEV	0x0	RW	**Controls When the Counter Counts**
	Selects whether the regular counter responds to up-count events, down-count events or both			
	Value	Mode		Description
	0	BOTH		Counts up on up-count and down on down-count events.
	1	UP		Only counts up on up-count events.
	2	DOWN		Only counts down on down-count events.
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	EDGE	0x0	RW	**Edge Select**
	Determines the polarity of the incoming edges. This bit used only in OVSSINGLE, EXTCLKSINGLE and OVSQUAD1X-4X modes.			
	Value	Mode		Description
	0	POS		Positive edges on the PCNTn_S0IN inputs are counted in OVSSINGLE mode. Does not invert PCNTn_S1IN input in OVSSINGLE and EXTCLKSINGLE modes
	1	NEG		Negative edges on the PCNTn_S0IN inputs are counted in OVSSINGLE mode. Inverts the PCNTn_S1IN input in OVSSINGLE and EXTCLKSINGLE modes
1	CNTDIR	0x0	RW	**Non-Quadrature Mode Counter Direction Co**
	The direction of the counter must be set in the OVSSINGLE and EXTCLKSINGLE modes. This bit is ignored in EXTCLKQUAD mode as the direction is automatically detected.			
	Value	Mode		Description
	0	UP		Up counter mode.
	1	DOWN		Down counter mode.

Bit	Name	Reset	Access	Description
0	S1CDIR	0x0	RW	**Count Direction Determined By S1** Allows S1 give the direction of counting when in the OVSSINGLE or EXTCLKSINGLE modes. When S1 is high, the count direction is given by CNTDIR. When S1 is low, the count direction is the opposite given by CNTDIR

29.5.6 PCNT_CMD - Command Register

Offset	Bit Position																													
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																					0x0	0x0	0x0	0x0				0x0		0x0	0x0	0x0
Access																					W	W	W	W				W(nB)		W(nB)	W(nB)	W(nB)
Name																					STOPAUXCNT	STOPCNT	STARTAUXCNT	STARTCNT				LCNTIM		AUXCNTRST	CNTRST	CORERST

Bit	Name	Reset	Access	Description
31:12	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
11	STOPAUXCNT	0x0	W	**Stop Aux Counter** Write a 1 to stop the aux counter
10	STOPCNT	0x0	W	**Stop Main Counter** Write a 1 to stop the main counter
9	STARTAUXCNT	0x0	W	**Start Aux Counter** Write a 1 to start the aux counter
8	STARTCNT	0x0	W	**Start Main Counter** Write a 1 to start the main counter
7:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	LCNTIM	0x0	W(nB)	**Load CNT Immediately** Load PCNTn_TOP into PCNTn_CNT on the next counter clock cycle.
3	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
2	AUXCNTRST	0x0	W(nB)	**AUXCNT Reset** The auxiliary counter, AUXCNT, is synchronously reset when this bit is set. This action clears the auxiliary counter to its reset value
1	CNTRST	0x0	W(nB)	**CNT Reset** The counter, CNT, is synchronously reset when this bit is set. This action clears the counter to its reset value
0	CORERST	0x0	W(nB)	**PCNT Clock Domain Reset** The PCNT clock domain logic is synchronously reset when this bit is set. This action clears the all the PCNT logic and registers to their reset value

29.5.7 PCNT_STATUS - Status Register

Offset	Bit Position																														
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												R	R	R	R	R
Name																												AUXCNTRUNNING	CNTRUNNING	PCNTLOCKSTATUS	TOPBV	DIR

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	AUXCNTRUNNING	0x0	R	**Aux Counter running status**
	Indicates the current status of PCNT auxiliary counter running. The counter status will continue to indicate as Running even when it is halted during debug assertion.			
3	CNTRUNNING	0x0	R	**Main Counter running status**
	Indicates the current status of PCNT main counter running. The counter status will continue to indicate as Running even when it is halted during debug assertion.			
2	PCNTLOCKSTATUS	0x0	R	**Lock Status**
	Indicates the current status of PCNT Lock			

Value	Mode	Description
0	UNLOCKED	PCNT registers are unlocked
1	LOCKED	PCNT registers are locked

Bit	Name	Reset	Access	Description
1	TOPBV	0x0	R	**TOP Buffer Valid**
	This indicates that PCNTn_TOPB contains valid data that has not been written to PCNTn_TOP. This bit is also cleared when PCNTn_TOP is written.			
0	DIR	0x0	R	**Current Counter Direction**
	Current direction status of the counter. This bit is valid in EXTCLKQUAD mode only.			

Value	Mode	Description
0	UP	Up counter mode (clockwise in EXTCLKQUAD mode with the EDGE bit in PCNTn_CTRL set to 0).
1	DOWN	Down counter mode.

29.5.8 PCNT_IF - Interrupt Flag Register

Offset	Bit Position																														
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												OQSTERR	AUXOF	DIRCNG	OF	UF

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	OQSTERR	0x0	RW	**Oversampling Quad State Err Int Flag**
	Set in the Oversampling Quadrature Mode when incorrect state transition occurs			
3	AUXOF	0x0	RW	**Auxiliary Overflow Interrupt Read Flag**
	Set when an Auxiliary CNT overflow occurs			
2	DIRCNG	0x0	RW	**Direction Change Detect Interrupt Flag**
	Set when the count direction changes. Set in EXTCLKQUAD mode only.			
1	OF	0x0	RW	**Overflow Interrupt Read Flag**
	Set when a CNT overflow occurs			
0	UF	0x0	RW	**Underflow Interrupt Read Flag**
	Set when a CNT underflow occurs			

29.5.9 PCNT_IEN - Interrupt Enable Register

Offset	Bit Position																															
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												OQSTERR	AUXOF	DIRCNG	OF	UF

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	OQSTERR	0x0	RW	**Oversampling Quad State Err Int Flag**
	Set to enable the OQSTERRIF Interrupt			
3	AUXOF	0x0	RW	**Auxiliary Overflow Interrupt Read Flag**
	Set to enable the AUXOFIF Interrupt			
2	DIRCNG	0x0	RW	**Direction Change Detect Interrupt Flag**
	Set to enable the DIRCNGIF Interrupt			
1	OF	0x0	RW	**Overflow Interrupt Read Flag**
	Set to enable the OFIF Interrupt			
0	UF	0x0	RW	**Underflow Interrupt Read Flag**
	Set to enable the UFIF Interrupt			

29.5.10 PCNT_CNT - Counter Value Register

Offset	Bit Position																															
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	R															
Name																	CNT															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	CNT	0x0	R	**Counter Value**
	Gives read access to the counter.			

29.5.11 PCNT_AUXCNT - Auxiliary Counter Value Register

Offset	Bit Position																													
0x028	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	R															
Name																	AUXCNT															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	AUXCNT	0x0	R	**Auxiliary Counter Value**
	Gives read access to the auxiliary counter.			

29.5.12 PCNT_TOP - Top Value Register

Offset	Bit Position																													
0x02C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0xFF															
Access																	RW															
Name																	TOP															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	TOP	0xFF	RW	**Counter Top Value**
	When counting down, this value is reloaded into PCNTn_CNT when counting past 0. When counting up, 0 is written to the PCNTn_CNT register when counting past this value.			

29.5.13 PCNT_TOPB - Counter Top Value Buffer Register

Offset	Bit Position																													
0x030	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0xFF															
Access																	RW															
Name																	TOPB															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	TOPB	0xFF	RW	**Counter Top Buffer Register**
	These bits hold the TOP buffer value. Hardware updates the TOP value from TOPB whenever there is Underflow or Overflow event on main counter. The update to TOP wont happen for Overflow event if PCNTn_CTRL.CNTEV = BOTH			

29.5.14 PCNT_OVSCTRL - Oversampling Control Register

Offset	Bit Position																													
0x034	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																				0x0					0x0							
Access																				RW					RW							
Name																				FLUTTERRM					FILTLEN							

Bit	Name	Reset	Access	Description
31:13	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
12	FLUTTERRM	0x0	RW	**Flutter Remove**
	When set, removes flutter from Quaddecoder inputs S0IN and S1IN. Available only in OVSQUAD1X-4X modes			
11:8	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
7:0	FILTLEN	0x0	RW	**Configure Filter Length for Inputs S0IN**
	Used only in OVSINGLE,OVSQUAD1X-4X modes. To use this first enable FILTEN in PCNTn_CFG register. Filter length = (FILTLEN + 5) EM23GRPACLK cycles			

29.5.15 PCNT_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																															
0x038	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												R	R	R	R	R
Name																												OVSCTRL	TOPB	TOP	CMD	CTRL

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	OVSCTRL	0x0	R	**OVSCTRL Register Busy**
	Set when the value written to OVSCTRL register is being synchronized			
3	TOPB	0x0	R	**TOPB Register Busy**
	Set when the value written to TOPB register is being synchronized			
2	TOP	0x0	R	**TOP Register Busy**
	Set when the value written to TOP register is being synchronized			
1	CMD	0x0	R	**CMD Register Busy**
	Set when the value written to CMD register being synchronized			
0	CTRL	0x0	R	**CTRL Register Busy**
	Set when the value written to CTRL register being synchronized			

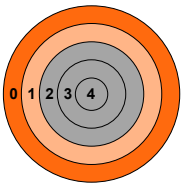
29.5.16 PCNT_LOCK - Configuration Lock Register

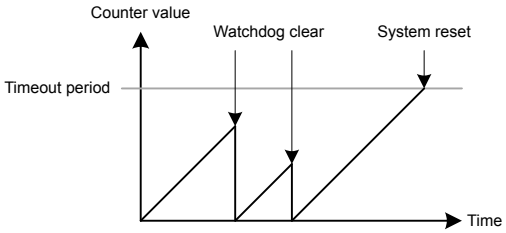
Offset	Bit Position																															
0x03C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0x0															
Access																	W															
Name																	PCNTLOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	PCNTLOCKKEY	0x0	W	**Configuration Lock Key**
	Write any other value than the unlock code to lock PCNT_CFG, PCNT_EN, PCNT_SWRST, PCNT_CMD, PCNT_CTRL, PCNT_OVSCTRL, PCNT_CNT, PCNT_TOP and PCNT_TOPB registers from editing. Write the unlock code to unlock.			

Value	Mode	Description
42976	UNLOCK	Write to unock PCNT lockable registers

30. WDOG - Watch Dog Timer





Quick Facts

What?

The WDOG resets the system in case of a fault condition, and can be enabled in all energy modes as long as the low frequency clock source is available.

Why?

If a software failure or external event renders the MCU unresponsive, a Watchdog timeout will reset the system to a known, safe state.

How?

An enabled Watchdog Timer implements a configurable timeout period. If the CPU fails to re-start the Watchdog Timer before it times out, a full system reset will be triggered. The Watchdog consumes insignificant power, and allows the device to remain safely in low-energy modes for up to 256 seconds at a time.

30.1 Introduction

The purpose of the watchdog timer is to generate a reset in case of a system failure to increase application reliability. The failure can be caused by a variety of events, such as an ESD pulse or a software failure.

30.2 Features

- Clock input from selectable oscillators
  - Internal 32 kHz LFRCO oscillator (WDOG0)
  - Internal 1 kHz ULFRCO oscillator (WDOG0)
  - External 32.768 kHz LFXO XTAL oscillator (WDOG0)
  - HCLK divided by 1024 (WDOG0 or WDOG1)
- Configurable timeout period from 9 to 256k watchdog clock cycles
- Individual selection to keep running or freeze when entering sleep modes
- Selection to keep running or freeze when entering debug mode
- Selection to block the CPU from entering Energy Mode 4
- Configurable warning interrupt at 25%, 50%, or 75% of the timeout period
- Configurable window interrupt at 12.5%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5% of the timeout period
- Timeout interrupt
- PRS as a watchdog clear
- Interrupt for the event where a PRS rising edge is absent before a software reset

30.3 Functional Description

The watchdog is enabled by setting the EN bit in WDOGn_EN. When enabled, the watchdog counts up to the period value configured through the PERSEL field in WDOGn_CFG. If the watchdog timer is not cleared to 0 (by writing a 1 to the CLEAR bit in WDOGn_CMD) before the period is reached, the chip is reset. If a timely clear command is issued, the timer starts counting up from 0 again. The watchdog can optionally be locked by writing anything other than UNLOCK code in WDOGn_LOCK. Once locked, it cannot be disabled or reconfigured by software.

Note: If the WDOG is configured to halt during a low energy mode and the energy mode is entered on the clock cycle before the specified timeout, a timeout event will occur upon the wake event.

When the EN bit in WDOGn_EN is cleared to 0, the watchdog counter is reset. Any pending interrupt flags will remain active until cleared.

30.3.1 Power Domains and Energy Modes

Individual WDOG instances are powered from different power domains in the device, which may impact clocking options and availability in different energy modes. See 30.3.2 Clock Source for more details on clocking. The power domain and energy mode availability for each instance are summarized in the following table.

Table 30.1. WDOG Power Domains

Instance	Power Domain	Energy Modes
WDOG0	PD0D	EM0, EM1
WDOG1	PD1	EM0, EM1

30.3.2 Clock Source

Four clock sources are available for use with WDOG0, through the CLKSEL field in CMU_WDOGnCLKCTRL. The selected oscillator source automatically starts when the watchdog is enabled. To prevent accidental change of the clock selection, CMU_WDOGLOCK can be written anything other than UNLOCK code. Also, respective oscillator has locks to prevent accidental disabling of oscillators. The PERSEL field in WDOGn_CFG is used to divide the selected watchdog clock, and the timeout for the watchdog timer can be calculated with the formula:

$$T_{TIMEOUT} = [2^{(PERSEL+3)} + 1] / f$$

where f is the frequency of the selected clock.

WDOG1 supports clocking only from HCLK/1024.

30.3.3 Debug Functionality

The watchdog timer can either keep running or be frozen when the device is halted by a debugger. This configuration is done through the DEBUGRUN bit in WDOGn_CFG. When code execution is resumed, the watchdog will continue counting where it left off.

30.3.4 Energy Mode Handling

The watchdog timer can be configured to either keep on running or freeze when entering low energy modes. The configuration is done individually for each WDOG in the EM1RUN bit in WDOGn_CFG. When the watchdog has been frozen and is re-entering an energy mode where it is running, the watchdog timer will continue counting where it left off. The watchdog does not run in EM4. If EM4BLOCK in WDOGn_CFG is set, the CPU will be prevented from entering EM4 by software request.

30.3.5 Warning Interrupt

The watchdog implements a warning interrupt which can be configured to occur at approximately 25%, 50%, or 75% of the timeout period through the WARNSEL field of the WDOGn_CFG register. This interrupt can be used to wake up the cpu for clearing the watchdog. The warning point for the watchdog timer can be calculated with the formula:

$$T_{WARNING} = [2^{(PERSEL+3)} +1]*WARNSEL / 4 / f$$

where f is the frequency of the selected clock.

When the watchdog is enabled, it is recommended to clear the watchdog before changing WARNSEL.

30.3.6 Window Interrupt

This interrupt occurs when the watchdog is cleared below a certain threshold. This threshold is given by the formula:

$$T_{WINDOW} = [2^{(PERSEL+3)} + 1]*WARNSEL / 8 / f$$

where f is the frequency of the selected clock.

This value will be approximately 12.5%, 25%, 37.5%, 50%, 62.5%, 75%, or 87.5% of the timeout value based on the WINSEL field of the WDOGn_CFG. Figure 30.1. WDOG Warning, Window, and Timeout illustrates the warning, the window, and the timeout interrupts. Also, it shows where the PRS rising edge needs to happen. The PRS edge detection feature is discussed later.

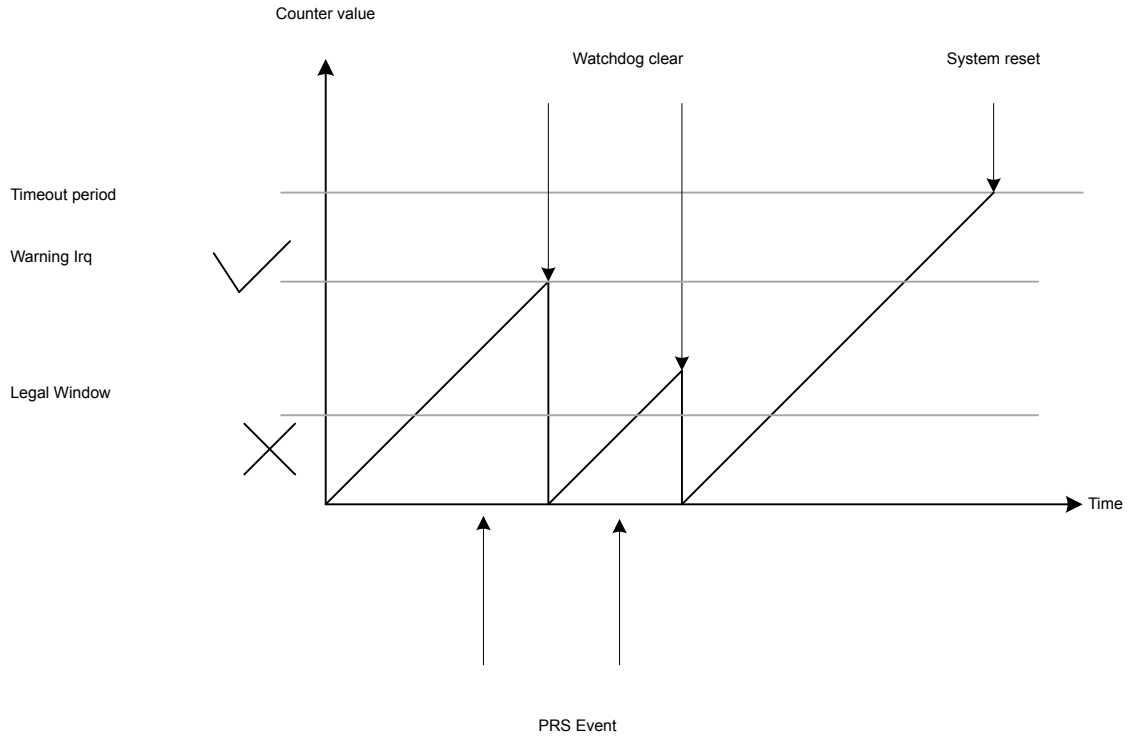


Figure 30.1. WDOG Warning, Window, and Timeout

30.3.7 PRS as Watchdog Clear

A PRS channel (selected by register PRS_CONSUMER_WDOGn_SRC0) can be used to clear the watchdog counter. To enable this feature, CLRSRC must be set to 1. Figure 30.2. PRS Clearing WDOG shows how the PRS channel takes over the WDOG clear function. Clearing the WDOG with the PRS is mutually exclusive of clearing the WDT by software.

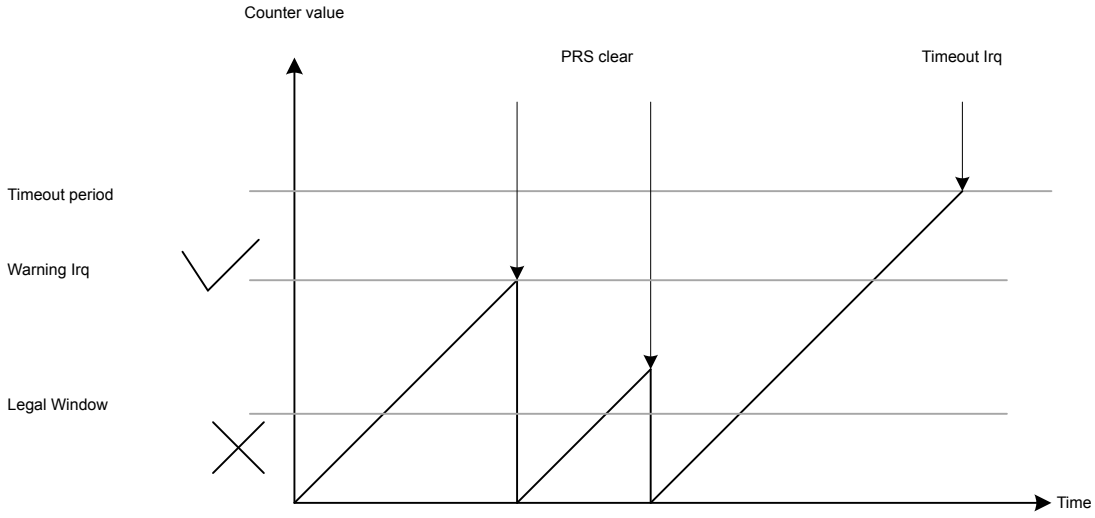


Figure 30.2. PRS Clearing WDOG

30.3.8 PRS Rising Edge Monitoring

PRS channels can be used to monitor multiple processes. The first and second channel are selected by PRS_CONSUMER_WDOGn_SRC0 and PRS_CONSUMER_WDOGn_SRC1, respectively. If enabled, every time the watch dog timer is cleared the PRS channels are checked and any channel which has not seen an event can trigger an interrupt.

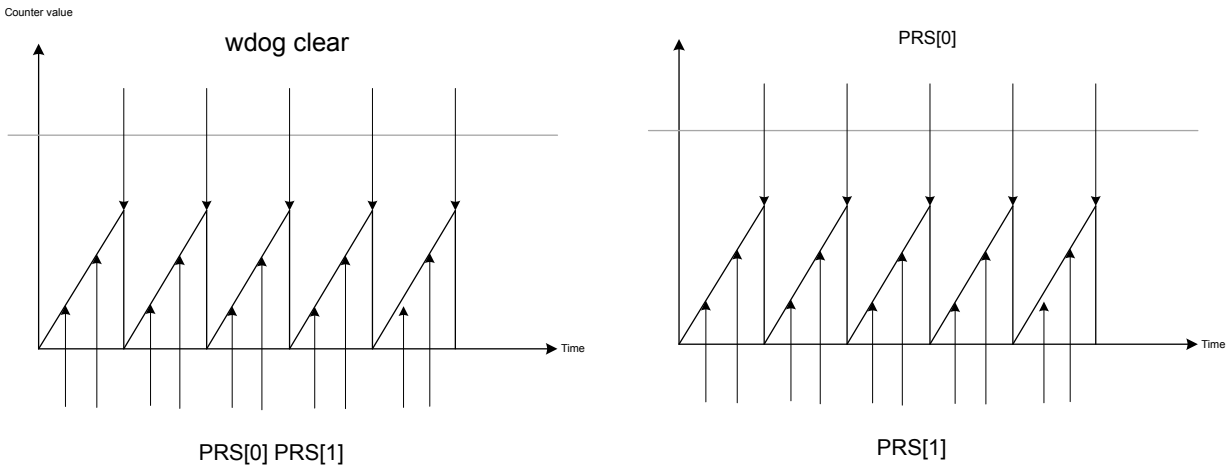


Figure 30.3. PRS Edge Monitoring in WDOG

30.4 WDOG Register Map

The offset register address is relative to the registers base address.

Offset	Name	Type	Description
0x000	WDOG_IPVERSION	R	IP Version Register
0x004	WDOG_EN	RW ENABLE	Enable Register
0x008	WDOG_CFG	RW CONFIG	Configuration Register
0x00C	WDOG_CMD	W LFSYNC	Command Register
0x014	WDOG_STATUS	RH	Status Register
0x018	WDOG_IF	RWH INTFLAG	Interrupt Flag Register
0x01C	WDOG_IEN	RW	Interrupt Enable Register
0x020	WDOG_LOCK	W	Lock Register
0x024	WDOG_SYNCBUSY	RH	Synchronization Busy Register
0x1000	WDOG_IPVERSION_SET	R	IP Version Register
0x1004	WDOG_EN_SET	RW ENABLE	Enable Register
0x1008	WDOG_CFG_SET	RW CONFIG	Configuration Register
0x100C	WDOG_CMD_SET	W LFSYNC	Command Register
0x1014	WDOG_STATUS_SET	RH	Status Register
0x1018	WDOG_IF_SET	RWH INTFLAG	Interrupt Flag Register
0x101C	WDOG_IEN_SET	RW	Interrupt Enable Register
0x1020	WDOG_LOCK_SET	W	Lock Register
0x1024	WDOG_SYNCBUSY_SET	RH	Synchronization Busy Register
0x2000	WDOG_IPVERSION_CLR	R	IP Version Register
0x2004	WDOG_EN_CLR	RW ENABLE	Enable Register
0x2008	WDOG_CFG_CLR	RW CONFIG	Configuration Register
0x200C	WDOG_CMD_CLR	W LFSYNC	Command Register
0x2014	WDOG_STATUS_CLR	RH	Status Register
0x2018	WDOG_IF_CLR	RWH INTFLAG	Interrupt Flag Register
0x201C	WDOG_IEN_CLR	RW	Interrupt Enable Register
0x2020	WDOG_LOCK_CLR	W	Lock Register
0x2024	WDOG_SYNCBUSY_CLR	RH	Synchronization Busy Register
0x3000	WDOG_IPVERSION_TGL	R	IP Version Register
0x3004	WDOG_EN_TGL	RW ENABLE	Enable Register
0x3008	WDOG_CFG_TGL	RW CONFIG	Configuration Register
0x300C	WDOG_CMD_TGL	W LFSYNC	Command Register
0x3014	WDOG_STATUS_TGL	RH	Status Register
0x3018	WDOG_IF_TGL	RWH INTFLAG	Interrupt Flag Register
0x301C	WDOG_IEN_TGL	RW	Interrupt Enable Register
0x3020	WDOG_LOCK_TGL	W	Lock Register
0x3024	WDOG_SYNCBUSY_TGL	RH	Synchronization Busy Register

30.5 WDOG Register Description

30.5.1 WDOG_IPVERSION - IP Version Register

Offset	Bit Position																													
0x000	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x1																															
Access	R																															
Name	IPVERSION																															

Bit	Name	Reset	Access	Description
31:0	IPVERSION	0x1	R	**IP Version**
	The read only IPVERSION field gives the version for this module. There may be minor software changes required for modules with different values of IPVERSION.			

30.5.2 WDOG_EN - Enable Register

Offset	Bit Position																													
0x004	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																															0x0	0x0
Access																															R	RW
Name																															DISABLING	EN

Bit	Name	Reset	Access	Description
31:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
1	DISABLING	0x0	R	**Disabling busy status**
	When EN is cleared, DISABLING status is set immediately, and cleared when disablement finishes. Disablement resets peripheral cores and not APB registers except hardware updated registers such as INTFLAGs and FIFO			
0	EN	0x0	RW	**Module Enable**
	The ENABLE bit enables the module. Software should write to CONFIG type registers before setting the ENABLE bit. Software should write to SYNC type registers only after setting the ENABLE bit.			

30.5.3 WDOG_CFG - Configuration Register

Offset	Bit Position																														
0x008	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset		0x0					0x0						0xF									0x0	0x0	0x0			0x0	0x0			0x0	0x0
Access		RW					RW						RW									RW	RW	RW			RW	RW			RW	RW
Name		WINSEL					WARNSEL						PERSEL									PRS1MISSRSTEN	PRS0MISSRSTEN	WDOGRSTDIS			DEBUGRUN	EM4BLOCK			EM1RUN	CLRSRC

Bit	Name	Reset	Access	Description
31	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
30:28	WINSEL	0x0	RW	**WDOG Illegal Window Select**
	Select WDOG illegal limit.			

Value	Mode	Description
0	DIS	Disabled.
1	SEL1	Window timeout is 12.5% of the Timeout.
2	SEL2	Window timeout is 25% of the Timeout.
3	SEL3	Window timeout is 37.5% of the Timeout.
4	SEL4	Window timeout is 50% of the Timeout.
5	SEL5	Window timeout is 62.5% of the Timeout.
6	SEL6	Window timeout is 75.5% of the Timeout.
7	SEL7	Window timeout is 87.5% of the Timeout.

Bit	Name	Reset	Access	Description
27:26	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
25:24	WARNSEL	0x0	RW	**WDOG Warning Period Select**
	Select WDOG warning timeout period.			

Value	Mode	Description
0	DIS	Disable
1	SEL1	Warning timeout is 25% of the Timeout.
2	SEL2	Warning timeout is 50% of the Timeout.
3	SEL3	Warning timeout is 75% of the Timeout.

Bit	Name	Reset	Access	Description
23:20	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
19:16	PERSEL	0xF	RW	**WDOG Timeout Period Select**
	Select WDOG timeout period.			

Value	Mode	Description
0	SEL0	Timeout period of 9 wdog cycles
1	SEL1	Timeout period of 17 wdog cycles

Bit	Name	Reset	Access	Description
	2	SEL2		Timeout period of 33 wdog cycles
	3	SEL3		Timeout period of 65 wdog cycles
	4	SEL4		Timeout period of 129 wdog cycles
	5	SEL5		Timeout period of 257 wdog cycles
	6	SEL6		Timeout period of 513 wdog cycles
	7	SEL7		Timeout period of 1k wdog cycles
	8	SEL8		Timeout period of 2k wdog cycles
	9	SEL9		Timeout period of 4k wdog cycles
	10	SEL10		Timeout period of 8k wdog cycles
	11	SEL11		Timeout period of 16k wdog cycles
	12	SEL12		Timeout period of 32k wdog cycles
	13	SEL13		Timeout period of 64k wdog cycles
	14	SEL14		Timeout period of 128k wdog cycles
	15	SEL15		Timeout period of 256k wdog cycles
15:11	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
10	PRS1MISSRSTEN	0x0	RW	**PRS Src1 Missing Event WDOG Reset**
	When set, a PRS Source 1 missing event will trigger a WDOG reset.			
9	PRS0MISSRSTEN	0x0	RW	**PRS Src0 Missing Event WDOG Reset**
	When set, a PRS Source 0 missing event will trigger a WDOG reset.			
8	WDOGRSTDIS	0x0	RW	**WDOG Reset Disable**
	Disable WDOG reset output.			
	Value	Mode		Description
	0	EN		A timeout will cause a WDOG reset
	1	DIS		A timeout will not cause a WDOG reset
7:6	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
5	DEBUGRUN	0x0	RW	**Debug Mode Run**
	Set to keep WDOG running in debug mode.			
	Value	Mode		Description
	0	DISABLE		WDOG timer is frozen in debug mode
	1	ENABLE		WDOG timer is running in debug mode
4	EM4BLOCK	0x0	RW	**EM4 Block**
	Set to disallow EM4 entry by software.			
	Value	Mode		Description
	0	DISABLE		EM4 can be entered by software. See EMU for detailed description.
	1	ENABLE		EM4 cannot be entered by software.
3:2	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		

Bit	Name	Reset	Access	Description
1	EM1RUN	0x0	RW	**EM1 Run**
	Set to keep WDOG running in EM1.			

Value	Mode	Description
0	DISABLE	WDOG timer is frozen in EM1.
1	ENABLE	WDOG timer is running in EM1.

Bit	Name	Reset	Access	Description
0	CLRSRC	0x0	RW	**WDOG Clear Source**
	Select WDOG clear source.			

Value	Mode	Description
0	SW	A write to the clear bit will clear the WDOG counter
1	PRSSRC0	A rising edge on the PRS Source 0 will clear the WDOG counter

30.5.4 WDOG_CMD - Command Register

Offset	Bit Position																													
0x00C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																W(nB)
Name																																CLEAR

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CLEAR	0x0	W(nB)	**WDOG Timer Clear**
	Clear WDOG timer. The bit must be written 4 WDOG cycles before the timeout.			

Value	Mode	Description
0	UNCHANGED	WDOG timer is unchanged.
1	CLEARED	WDOG timer is cleared to 0.

30.5.5 WDOG_STATUS - Status Register

Offset	Bit Position																															
0x014	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset	0x0																															
Access	R																															
Name	LOCK																															

Bit	Name	Reset	Access	Description
31	LOCK	0x0	R	**WDOG Configuration Lock Status**
	Status of all lockable WDOG registers.			

Value	Mode	Description
0	UNLOCKED	All WDOG lockable registers are unlocked.
1	LOCKED	All WDOG lockable registers are locked.

30:0	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*

30.5.6 WDOG_IF - Interrupt Flag Register

Offset	Bit Position																															
0x018	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												PEM1	PEM0	WIN	WARN	TOUT

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PEM1	0x0	RW	**PRS Src1 Event Missing Interrupt Flag**
	Set when a WDOG clear happens before a prs event has been detected on PRS Source one.			
3	PEM0	0x0	RW	**PRS Src0 Event Missing Interrupt Flag**
	Set when a WDOG clear happens before a prs event has been detected on PRS Source zero.			
2	WIN	0x0	RW	**WDOG Window Interrupt Flag**
	Set when a WDOG clear happens below the window limit value.			
1	WARN	0x0	RW	**WDOG Warning Timeout Interrupt Flag**
	Set when a WDOG warning timeout has occurred.			
0	TOUT	0x0	RW	**WDOG Timeout Interrupt Flag**
	Set when a WDOG timeout has occurred.			

30.5.7 WDOG_IEN - Interrupt Enable Register

Offset	Bit Position																													
0x01C	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																												0x0	0x0	0x0	0x0	0x0
Access																												RW	RW	RW	RW	RW
Name																												PEM1	PEM0	WIN	WARN	TOUT

Bit	Name	Reset	Access	Description
31:5	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
4	PEM1	0x0	RW	**PRS Src1 Event Missing Interrupt Enable**
	Enable/disable the PEM1 interrupt.			
3	PEM0	0x0	RW	**PRS Src0 Event Missing Interrupt Enable**
	Enable/disable the PEM0 interrupt.			
2	WIN	0x0	RW	**WDOG Window Interrupt Enable**
	Enable/disable the WIN interrupt.			
1	WARN	0x0	RW	**WDOG Warning Timeout Interrupt Enable**
	Enable/disable the WARN interrupt.			
0	TOUT	0x0	RW	**WDOG Timeout Interrupt Enable**
	Enable/disable the TOUT interrupt.			

30.5.8 WDOG_LOCK - Lock Register

Offset	Bit Position																													
0x020	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																	0xABE8															
Access																	W															
Name																	LOCKKEY															

Bit	Name	Reset	Access	Description
31:16	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
15:0	LOCKKEY	0xABE8	W	**WDOG Configuration Lock**
	Write any other value than the unlock code to lock WDOG_EN, WDOD_CFG registers from editing. Write the unlock code to unlock.			

Value	Mode	Description
0	LOCK	Lock WDOG lockable registers
44008	UNLOCK	Unlock WDOG lockable registers

30.5.9 WDOG_SYNCBUSY - Synchronization Busy Register

Offset	Bit Position																													
0x024	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reset																																0x0
Access																																R
Name																																CMD

Bit	Name	Reset	Access	Description
31:1	*Reserved*	*To ensure compatibility with future devices, always write Reserved bits to their reset value, unless otherwise stated. More information in 1.2 Conventions*		
0	CMD	0x0	R	**Sync Busy for Cmd Register**
	CMD bitfield sync is busy when set.			

31. Revision History

Revision 1.0

June, 2026

- Removed references to Bluetooth Mesh throughout the document.
- Updated references to "ZigBee" to "Zigbee" throughout the document.
- Consolidated the marking of High Speed interrupts in 3.3.3 Interrupt Request Lines (IRQ).
- Fixed incorrect description of bits in ETAMPDET_CFG register in 12.5 ETAMPDET Register Description.
- Updated available features/peripherals in 13.3.1 Energy Modes.
- Clarified acqusition time in 23.3.3 Acquisition and Conversion Timing.
- Inserted GPIO locations for 25.3.9.1 JTAG Debug Connection and 25.3.9.2 Serial Wire Debug Connection.
- Made minor style updates throughout the document.

Revision 0.5

August, 2025

- Added bit and register names where placeholders existed in SYSRTC, SMU, PIXELRZ and LEDDRV modules.
- 13.10 Temperature Sensor : Removed section detailing "Linearization, Offset Correction, and Calibration".
- Filled in TBD placeholders with determined information.
- 9.3 SOCPLL - SoC Phase-Locked-Loop: Corrected frequency of operation to show 150 MHz (145 MHz open-loop) instead of 100-150 MHz range.
- 9.6 LFXO - Low-Frequency Crystal Oscillator: Corrected descriptions for modes in LFXO_CFG.MODE bitfield.

Revision 0.4

April, 2025

- SYXO replaced with HFXO in register and bit names to adhere to naming conventions.
- Updated all registers to silicon revision B versions.
- 3.3.3 Interrupt Request Lines (IRQ): Added KSU interrupt vectors.
- Table 5.3. GP Fragment: Added HFRCOEM2340MHZ calibration slot.
- 10.5.30 SMU_ESAURTYPES - ESAU Region Types: Removed ESAUTZPOLARITY bit.
- 13.12.3 EMU_FLREG - FLREG Control Register: Added OVROVERRIDEEN and OVRENLP bits.
- 14.3.3.1 Producer Details: Added KSU signals.
- 23.3.3.2.1 Conversion Timing Examples: Updated conversion timing examples and added example of using REPDELAY.
- 23.3.5 Input and Configuration Selection: Added note about VSS paddle being the ground reference for single-ended conversions.
- 27.3.3 Charge Pump Operation: Corrected nominal LEDDRV charge pump capacitor value.

Revision 0.3

September, 2024

- Documented IOVDD1BOD path as alternative to IOVDD0BOD.
- Table 10.2. Secure Access Table: Corrected Result in row 3 to "Success".
- Removed QSPI and FLPLL clock controls in CMU as they are unavailable to user code.

Revision 0.2

August, 2024

- Normalized and corrected references to registers, bitfields, modes, pins, and signals throughout document.
- 23.3.7.3 Digital Accumulation and Averaging: Updated to reflect differences in output word formatting between revision A and revision B silicon when ALIGNMENT = RIGHT16.
- Table 4.1. Host Core Address Map: Fixed descriptions to refer to instruction and data bus regions rather than fixed 'cacheable' and 'non-cacheable' aliases.
- 4.2.4 Peripherals: Updated peripheral map to show default secure vs. non-secure regions.
- 26.3.7.2 SYNC Descriptor Structure: Updated to correct bitfield positions of MATCHVAL and MATCHEN

Revision 0.1

May, 2024

Initial Release.

Appendix A Abbreviations

This section lists abbreviations which may be used in this document.

Table A.1. Abbreviations

Abbreviation	Description
ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
AFC	Automatic Frequency Control
AGC	Automatic Gain Control
AHB	Advanced High-Speed Bus
AMBA	Advanced Microcontroller Bus Architecture (ARM defined system of bus infrastructure and protocols such as AHB, AXI, APB)
APB	Advanced Peripheral Bus
APC	Automatic Power Control
ASK	Amplitude Shift Keying
AXI	Advanced eXtensible Interface
BLE	Bluetooth Low Energy
BR	Baud Rate
BT	Bandwidth Time product
BUFC	Buffer Controller
BW	Bandwidth
CBC	Cipher Block Chaining (AES mode of operation)
CBC-MAC	Cipher Block Chaining - Message Authentication Code (AES mode of operation)
CC	Compare / Capture
CCA	Clear Channel Assessment
CFB	Cipher Feedback (AES mode of operation)
CHF	Channel Filter
CLK	Clock
CM33	ARM Cortex-M33
CMD	Command
CMU	Clock Management Unit
CRC	Cyclic Redundancy Check
CTR	Counter mode (AES mode of operation)
CTRL	Control
DBG	Debug
DC	Direct Current
DEC	Decimator
DEMOD	Demodulator
DSA	Detection of Signal Arrival
DSSS	Direct Sequence Spread Spectrum
DMA	Direct Memory Access

Abbreviation	Description
ECB	Electronic Code Book (AES mode of operation)
ECC	Ecliptic Curve Cryptography
ECC	Error Checking Code
EM	Energy Mode
EMU	Energy Management Unit
EXTMEM	External Memory Interface
FEC	Forward Error Correction
FIR	Finite Impulse Response
FRC	Frame Controller
FSK	Frequency Shift Keying
FSM	Finite State Machine
GFSK	Gaussian Frequency Shift Keying
GMSK	Gaussian Minimum Shift Keying
GPIO	General Purpose Input / Output
HFRCO	High Frequency RC Oscillator
HFXO	High Frequency Crystal Oscillator
HW	Hardware
Hz	Hertz
IF	Intermediate Frequency
IFADC	Intermediate Frequency Analog to Digital Converter
ISR	Interrupt Service Routine
KB	Kilobyte (1024 bytes)
LFRCO	Low Frequency RC Oscillator
LFXO	Low Frequency Crystal Oscillator
LNA	Low Noise Amplifier
LO	Local Oscillator
MB	Megbyte (1024 kilobytes)
MOD	Modulator
MODEM	Modulator and Demodulator
MSK	Minimum Shift Keying
NRZ	Non Return to Zero
NVIC	Nested Vector Interrupt Controller
NVM	Non-Volatile Memory
OFB	Output Feedback Mode (AES mode of operation)
OOK	On Off Keying
OQPSK	Offset Quadrature Phase Shift Keying
PA	Power Amplifier
PD	Power Domain
PHY	Physical Layer
PROTIMER	Protocol Timer
PRS	Peripheral Reflex System

Abbreviation	Description
PWM	Pulse Width Modulation
RAC	Radio Controller
RAM	Random Access Memory
RF	Radio Frequency
RMU	Reset Management Unit
RSM	Radio State Machine
RSSI	Received Signal Strength Indicator
RTC	Real Time Counter
RX	Receive
SEQ	Radio Sequencer
SPI	Serial Peripheral Interface
SW	Software
SYNTH	Synthesizer
TX	Transmit
XTAL	Crystal

SILICON LABS
CONNECTED INTELLIGENCE

The Leading Innovator in Low-Power Wireless.

IoT Portfolio
silabs.com/products

Simplicity Studio
silabs.com/simplicity

Quality
silabs.com/quality

Support & Community
silabs.com/community

Disclaimer

Silicon Labs provides customers with data sheets, application notes and product briefs ("Product Documentation") that are intended to present current, accurate and in-depth information about Silicon Labs' integrated circuits, peripherals, modules and other products ("Products"). Silicon Labs may also provide reference designs, schematics, design files, software, tools, documentation, consultation and other customer support materials ("Support Materials") that are intended solely to assist customers in developing their own modules, boards or other applications or products that include one or more Silicon Labs Products and may not be used for any other purposes.

Characterization data, device specifications, memory sizes, memory addresses and typical parameters can and do vary in different applications and are provided for illustrative purposes only, are subject to change without notice, and should not be relied upon without independent verification. Product firmware may be updated during the manufacturing process for security or reliability purposes.

Product Documentation does not imply or expressly grant any license to design or fabricate integrated circuits or modules. Customers are solely responsible for the design and distribution of their products including the acquisition of licenses to any necessary intellectual property rights.

Product Documentation and Support Materials are not authorized for use in or in connection with FDA Class III devices or military, automotive or aviation applications and shall in no circumstances be used in weapons of mass destruction or systems capable of delivering such weapons.

SILICON LABS DOES NOT WARRANT THAT PRODUCT DOCUMENTATION OR SUPPORT MATERIALS WILL MEET YOUR REQUIREMENT OR THAT THEIR OPERATION WILL BE UNINTERRUPTED OR THAT ANY SOFTWARE WILL BE ERROR-FREE. PRODUCT DOCUMENTATION AND SUPPORT MATERIALS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." SILICON LABS MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, REGARDING THE ACCURACY, COMPLETENESS, RELIABILITY, OR SUITABILITY OF PRODUCT DOCUMENTATION OR SUPPORT MATERIALS, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SILICON LABS DISCLAIMS ALL IMPLIED WARRANTIES.

Trademark Notice

Silicon Laboratories®, Silicon Labs®, Silabs®, and the Silicon Labs S-logo as well as other product or service names used herein are trademarks or registered trademarks of Silicon Laboratories Inc. For more information, please visit: Silicon Labs Trademark Use Guidelines - Silicon Labs. Arm®, Cortex® and Keil® are trademarks or registered trademarks of Arm Holdings or Arm Limited. All other third-party products or brand names mentioned herein are trademarks of their respective holders.

SILICON LABS CONNECTED INTELLIGENCE

400 West Cesar Chavez
Austin, TX 78701 | USA

www.silabs.com