ambiq

DATASHEET

# Apollo SoC

Ultra-low Power Apollo SoC Family

Doc. ID: DS-A1-1p2

Doc. Revision: 1.2, November 2023

## Features

Ultra-low supply current:
- EEMBC ULPBench score of 377
- EEMBC ULPBench score of 377
- 35 µA/MHz executing from FLASH at 3.3 V
- 143 nA deep sleep mode at 3.3 V
- 419 nA deep sleep mode with XTAL-assisted RTC at 3.3 V

High-performance ARM Cortex-M4 Processor
- Up to 24 MHz clock frequency
- Floating point unit
- Memory protection unit
- Wake-up interrupt controller with 12 interrupts

Ultra-low power memory:

- Up to 512 KB of flash memory for code/data
- Up to 64 KB of low leakage RAM for code/data

Ultra-low power interface for on- and off-chip sensors:
- 10 bit, 13-channel, up to 800 kSps ADC
- Temperature sensor with +/- 4ºC accuracy after calibration

Flexible serial peripherals:
- $I^2C$/SPI masters for communication with sensors, radios, and other peripherals
- $I^2C$/SPI slave for host communications
- UART for communication with peripherals and legacy devices

Rich set of clock sources:
- 32.768 kHz XTAL oscillator
- Low frequency RC oscillator – 1.024 kHz
- High frequency RC oscillator – 24 MHz
- RTC based on Ambiq's AM08X5/18X5 families

Wide operating range: 1.8-3.8 V, –40 to 85°C

Compact package:
- 2.49 x 2.90 mm 41-pin CSP with 27 GPIO
- 4.5 x 4.5 mm 64-pin BGA with 50 GPIO

## Applications

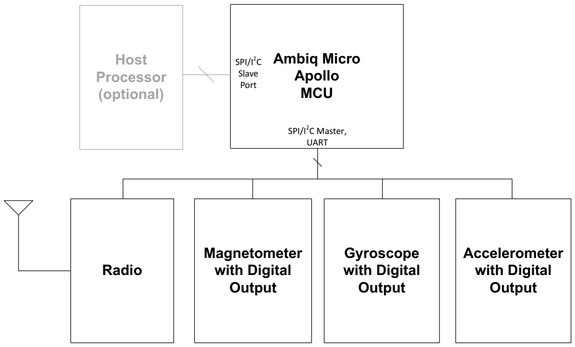
- Wearable electronics
- Wireless sensors
- Activity and fitness monitors
- Consumer medical devices

## Description

The Apollo SoC family is an ultra-low power, highly integrated microcontroller designed for battery-powered devices including wearable electronics, activity & fitness monitors, and wireless sensors. By combining ultra-low power sensor conversion electronics with the powerful ARM Cortex-M4F processor, the Apollo SoC enables complex sensor processing tasks to be completed with unprecedented battery life. Weeks, months, and years of battery life are achievable while doing complex context detection, gesture recognition, and activity monitoring. The Apollo SoC takes full advantage of Ambiq Micro's patented Subthreshold Power Optimized Technology (SPOT) Platform, setting a new industry benchmark in low power design.

The Apollo SoC also integrates up to 512 KB of flash memory and 64 KB of RAM to accommodate radio and sensor overhead while still leaving plenty of space for application code. This microcontroller also includes a serial master and UART port for communicating with radios and sensors including accelerometers, gyroscopes, and magnetometers.

Typical Sensor Application Circuit for the Apollo MCU


Host Processor (optional)
SPI/I$^2$C Slave Port
Ambiq Micro Apollo MCU
SPI/I$^2$C Master, UART
Radio
Magnetometer with Digital Output
Gyroscope with Digital Output
Accelerometer with Digital Output

## Table of Content

1. Apollo SoC Package Pins ........ 24
1.1 Pin Configuration ........ 24
1.2 Pin Connections ........ 26
2. System Core ........ 34
3. MCU Core Details ........ 36
3.1 Functional Overview ........ 36
3.2 Interrupts ........ 36
3.3 Memory Map ........ 38
3.4 Memory Protection Unit (MPU) ........ 40
3.5 System Buses ........ 41
3.6 Power Management ........ 41
3.6.1 Cortex-M4 Power Modes ........ 41
3.7 Debug Interfaces ........ 43
3.7.1 Instrumentation Trace Macrocell (ITM) ........ 43
3.7.2 Trace Port Interface Unit (TPIU) ........ 43
3.7.3 Faulting Address Trapping Hardware ........ 43
3.8 ITM Registers ........ 44
3.8.1 Register Memory Map ........ 44
3.8.2 ITM Registers ........ 45
3.9 MCUCTRL Registers ........ 71
3.9.1 Register Memory Map ........ 71
3.9.2 MCUCTRL Registers ........ 72
3.10 Memory Subsystem ........ 84
3.10.1 Features ........ 84
3.10.2 Functional Overview ........ 85
4. I2C/SPI Master Module ........ 86
4.1 Functional Overview ........ 86
4.2 Interface Clock Generation ........ 87
4.3 Command Operation ........ 87
4.4 FIFO ........ 88
4.5 I2C Interface ........ 88
4.5.1 Bus Not Busy ........ 88
4.5.2 Start Data Transfer ........ 88
4.5.3 Stop Data Transfer ........ 89
4.5.4 Data Valid ........ 89
4.5.5 Acknowledge ........ 89
4.5.6 I2C Slave Addressing ........ 89
4.5.7 I2C Offset Address Transmission ........ 90
4.5.8 I2C Normal Write Operation ........ 90
4.5.9 I2C Normal Read Operation ........ 91
4.5.10 I2C Raw Write Operation ........ 91
4.5.11 I2C Raw Read Operation ........ 91
4.5.12 Holding the Interface with CONT ........ 92
4.5.13 I2C Multi-master Arbitration ........ 92

4.6 SPI Operations ... 92
4.6.1 SPI Configuration ... 92
4.6.2 SPI Slave Addressing ... 93
4.6.3 SPI Normal Write ... 93
4.6.4 SPI Normal Read ... 93
4.6.5 SPI Raw Write ... 94
4.6.6 SPI Raw Read ... 94
4.6.7 Complex SPI Operations ... 95
4.6.8 SPI Polarity and Phase ... 95
4.7 Bit Orientation ... 96
4.8 Minimizing Power ... 96
4.9 IOMSTR Registers ... 97
4.9.1 Register Memory Map ... 97
4.9.2 IOMSTR Registers ... 98
5. I2C/SPI Slave Module ... 109
5.1 Functional Overview ... 109
5.2 Local RAM Allocation ... 110
5.3 Direct Area Functions ... 111
5.4 FIFO Area Functions ... 112
5.5 Rearranging the FIFO ... 114
5.6 Interface Interrupts ... 114
5.7 Host Address Space and Registers ... 115
5.8 I2C Interface ... 115
5.8.1 Bus Not Busy ... 116
5.8.2 Start Data Transfer ... 116
5.8.3 Stop Data Transfer ... 116
5.8.4 Data Valid ... 116
5.8.5 Acknowledge ... 116
5.8.6 Address Operation ... 117
5.8.7 Offset Address Transmission ... 117
5.8.8 Write Operation ... 118
5.8.9 Read Operation ... 118
5.8.10 General Address Detection ... 119
5.9 SPI Interface ... 119
5.9.1 Write Operation ... 119
5.9.2 Read Operation ... 120
5.9.3 Configuring 3-wire vs. 4-wire SPI Mode ... 120
5.9.4 SPI Polarity and Phase ... 120
5.10 Bit Orientation ... 121
5.11 Wakeup Using the I2C/SPI Slave ... 121
5.12 IOSLAVE Registers ... 122
5.12.1 Register Memory Map ... 122
5.12.2 IOSLAVE Registers ... 123
5.13 Host Side Address Space and Register ... 135
5.13.1 Host Address Space and Registers ... 135
6. GPIO and Pad Configuration Module ... 140

6.1 Functional Overview ........ 140
6.2 Pad Configuration Functions ........ 140
6.3 General Purpose I/O (GPIO) Functions ........ 144
6.3.1 Configuring the GPIO Functions ........ 144
6.3.2 Reading from a GPIO Pad ........ 144
6.3.3 Writing to a GPIO Pad ........ 144
6.3.4 GPIO Interrupts ........ 144
6.4 Pad Connection Summary ........ 145
6.4.1 Output Selection ........ 145
6.4.2 Output Control ........ 145
6.4.3 Input Control ........ 147
6.4.4 Pull-up Control ........ 147
6.4.5 Analog Pad Configuration ........ 147
6.5 Module-specific Pad Configuration ........ 147
6.5.1 Implementing IO Master Connections ........ 147
6.5.2 Implementing IO Slave Connections ........ 151
6.5.3 Implementing Counter/Timer Connections ........ 154
6.5.4 Implementing UART Connections ........ 155
6.5.5 Implementing GPIO Connections ........ 157
6.5.6 Implementing CLKOUT Connections ........ 157
6.5.7 Implementing ADC Connections ........ 157
6.5.8 Implementing Voltage Comparator Connections ........ 159
6.5.9 Implementing the Software Debug Port Connections ........ 159
6.6 GPIO Registers ........ 161
6.6.1 Register Memory Map ........ 161
6.6.2 GPIO Registers ........ 162
7. Clock Generator and Real Time Clock Module ........ 228
7.1 Clock Generator ........ 228
7.1.1 Functional Overview ........ 228
7.1.2 Low Frequency RC Oscillator (LFRC) ........ 229
7.1.3 High Precision XT Oscillator (XT) ........ 229
7.1.4 High Frequency RC Oscillator (HFRC) ........ 230
7.1.5 HFRC Auto-adjustment ........ 230
7.1.6 Frequency Measurement ........ 231
7.1.7 Generating 100 Hz ........ 231
7.1.8 XT Oscillator Failure Detection ........ 232
7.1.9 HFRC Stability Delay ........ 232
7.2 CLKGEN Registers ........ 232
7.2.1 Register Memory Map ........ 233
7.2.2 CLKGEN Registers ........ 233
7.3 Real Time Clock ........ 246
7.3.1 RTC Functional Overview ........ 246
7.3.2 Calendar Counters ........ 246
7.3.3 Calendar Counter Reads ........ 246
7.3.4 Alarms ........ 247
7.3.5 12/24 Hour Mode ........ 247

7.3.6 Century Control and Leap Year Management ........ 247
7.3.7 Weekday Function ........ 248
7.4 RTC Registers ........ 248
7.4.1 Register Memory Map ........ 248
7.4.2 RTC Registers ........ 248
8. Counter/Timer Module (CTIMER) ........ 256
8.1 Functional Overview ........ 256
8.2 Counter/Timer Functions ........ 256
8.2.1 Single Count (FN = 0) ........ 257
8.2.2 Repeated Count (FN = 1) ........ 257
8.2.3 Single Pulse (FN = 2) ........ 258
8.2.4 Repeated Pulse (FN = 3) ........ 258
8.2.5 Continuous (FN = 4) ........ 259
8.3 Creating 32-bit Counters ........ 260
8.4 Measuring Buck Converter Charge Insertion ........ 260
8.5 Generating the Sample Rate for the ADC ........ 260
8.6 CTIMER Registers ........ 260
8.6.1 Register Memory Map ........ 261
8.6.2 CTIMER Registers ........ 262
9. Watchdog Timer Module ........ 284
9.1 Functional Overview ........ 284
9.2 WDT Registers ........ 285
9.2.1 Register Memory Map ........ 285
9.2.2 WDT Registers ........ 285
10. Reset Generator Module ........ 290
10.1 Functional Overview ........ 290
10.2 External Reset Pin ........ 290
10.3 Power-on Event ........ 291
10.4 Brown-out Event ........ 291
10.5 Software Reset ........ 291
10.6 Watchdog Reset ........ 291
10.7 RSTGEN Registers ........ 292
10.7.1 Register Memory Map ........ 292
10.7.2 RSTGEN Registers ........ 292
10.7.3 CFG Register ........ 292
11. UART Module ........ 299
11.1 Features ........ 299
11.2 Functional Overview ........ 299
11.3 Enabling and Selecting the UART Clock ........ 300
11.4 Configuration ........ 300
11.5 Transmit FIFO and Receive FIFO ........ 300
11.6 UART Registers ........ 301
11.6.1 Register Memory Map ........ 301
11.6.2 UART Registers ........ 301
12. ADC and Temperature Sensor Module ........ 313
12.1 Features ........ 313

12.2 Functional Overview .... 314
12.2.1 Clock Source and Dividers .... 314
12.2.2 12 Channel Analog Mux .... 314
12.2.3 Triggering and Trigger Sources .... 314
12.3 Voltage Reference Sources .... 315
12.3.1 Eight Automatically Managed Conversion Slots .... 315
12.3.2 Automatic Sample Accumulation and Scaling .... 316
12.3.3 Eight Entry Result FIFO .... 318
12.3.4 Window Comparator .... 318
12.4 Operating Modes and the Mode Controller .... 319
12.4.1 Single Mode .... 320
12.4.2 Repeat Mode .... 321
12.4.3 Low Power Modes .... 321
12.5 Interrupts .... 321
12.6 Voltage Divider and Switchable Battery Load .... 322
12.7 ADC Registers .... 323
12.7.1 Register Memory Map .... 324
12.7.2 ADC Registers .... 324
13. Voltage Comparator Module .... 346
13.1 Functional Overview .... 346
13.2 VCOMP Registers .... 347
13.2.1 Register Memory Map .... 347
13.2.2 VCOMP Registers .... 348
14. Voltage Regulator Module .... 354
14.1 Functional Overview .... 354
15. Electrical Characteristics .... 355
15.1 Absolute Maximum Ratings .... 356
15.2 Recommended Operating Conditions .... 357
15.3 Current Consumption .... 358
15.4 Power Mode Transitions .... 359
15.5 Clocks/Oscillators .... 360
15.6 Analog-to-Digital Converter (ADC) .... 361
15.7 Buck Converter .... 363
15.8 Power-On RESET (POR) and Brown-Out Detector (BOD) .... 364
15.9 Resets .... 365
15.10 Voltage Comparator (VCOMP) .... 366
15.11 Internal DAC Reference for VCOMP .... 367
15.12 Inter-Integrated Circuit (I2C) Interface .... 368
15.13 Serial Peripheral Interface (SPI) Master Interface .... 369
15.14 Serial Peripheral Interface (SPI) Slave Interface .... 371
15.15 Universal Asynchronous Receiver/Transmitter (UART) .... 374
15.16 Counter/Timer (CTIMER) .... 375
15.17 Flash Memory .... 376
15.18 General Purpose Input/Output (GPIO) .... 377
15.19 Serial Wire Debug (SWD) .... 379
16. Package Mechanical Information .... 380

16.1 BGA Package ........ 380
16.1.1 PCB land pattern and solder stencil ........ 382
16.2 CSP Package ........ 383
16.3 Reflow Profile ........ 384
17. Ordering Information ........ 395
18. Document Revision History ........ 396

## List of Figures

Figure 1. CSP Pin Configuration Diagram (Top View — Balls on Bottom) ............................ 24
Figure 2. BGA Pin Configuration Diagram (Top View — Balls on Bottom) ........................... 25
Figure 3. Block Diagram for the Ultra-Low Power Apollo SoC .............................................. 34
Figure 4. Block Diagram for Flash and OTP Subsystem ........................................................ 84
Figure 5. Block Diagram for the I2C/SPI Master Module ...................................................... 86
Figure 6. I2C/SPI Master Clock Generation ........................................................................... 87
Figure 7. Basic I2C Conditions ............................................................................................... 88
Figure 8. I2C Acknowledge ..................................................................................................... 89
Figure 9. I2C 7-bit Address Operation ................................................................................... 89
Figure 10. I2C 10-bit Address Operation ............................................................................... 90
Figure 11. I2C Offset Address Transmission ......................................................................... 90
Figure 12. I2C Normal Write Operation ................................................................................. 90
Figure 13. I2C Normal Read Operation .................................................................................. 91
Figure 14. I2C Raw Write Operation ...................................................................................... 91
Figure 15. I2C Raw Read Operation ....................................................................................... 92
Figure 16. SPI Normal Write Operation (Single-byte Offset Address) .................................. 93
Figure 17. SPI Normal Read Operation .................................................................................. 94
Figure 18. SPI Raw Write Operation ...................................................................................... 94
Figure 19. SPI Raw Read Operation ....................................................................................... 94
Figure 20. SPI Combined Operation ....................................................................................... 95
Figure 21. SPI CPOL and CPHA ............................................................................................ 95
Figure 22. Block diagram for the I2C/SPI Slave Module ...................................................... 109
Figure 23. I2C/SPI Slave Module LRAM Addressing ........................................................... 110
Figure 24. I2C/SPI Slave Module FIFO ................................................................................. 113
Figure 25. Basic I2C Conditions ............................................................................................. 116
Figure 26. I2C Acknowledge ................................................................................................... 117
Figure 27. I2C 7-bit Address Operation ................................................................................. 117
Figure 28. I2C 10-bit Address Operation ............................................................................... 117
Figure 29. I2C Offset Address Transmission ......................................................................... 118
Figure 30. I2C Write Operation .............................................................................................. 118
Figure 31. I2C Read Operation ............................................................................................... 118
Figure 32. SPI Write Operation .............................................................................................. 119
Figure 33. SPI Read Operation ............................................................................................... 120
Figure 34. SPI CPOL and CPHA ............................................................................................ 120
Figure 35. Block diagram for the General Purpose I/O (GPIO) Module ................................ 140
Figure 36. Pad Connection Details ......................................................................................... 146
Figure 37. Block diagram for the Clock Generator and Real Time Clock Module ................. 228
Figure 38. Block diagram for the Real Time Clock Module .................................................. 246
Figure 39. Block Diagram for One Counter/Timer Pair ......................................................... 256
Figure 40. Counter/Timer Operation, FN = 0 ......................................................................... 257
Figure 41. Counter/Timer Operation, FN = 1 ......................................................................... 258
Figure 42. Counter/Timer Operation, FN = 2 ......................................................................... 258
Figure 43. Counter/Timer Operation, FN = 3 ......................................................................... 259
Figure 44. Counter/Timer Operation, FN = 4 ......................................................................... 260

Figure 45. Block Diagram for the Watchdog Timer Module .................................................. 284
Figure 46. Block diagram for the Reset Generator Module ..................................................... 290
Figure 47. Block diagram of circuitry for Reset pin ............................................................... 291
Figure 48. Block Diagram for the UART Module ..................................................................... 299
Figure 49. Block Diagram for ADC and Temperature Sensor ................................................ 313
Figure 50. Scan Flowchart ........................................................................................................ 320
Figure 51. ADC State Diagram .................................................................................................. 322
Figure 52. Switchable Battery Load .......................................................................................... 323
Figure 53. Block diagram for the Voltage Comparator Module .............................................. 346
Figure 54. I2C Timing ................................................................................................................ 368
Figure 55. SPI Master Mode, Phase = 0 .................................................................................. 369
Figure 56. SPI Master Mode, Phase = 1 .................................................................................. 370
Figure 57. SPI Slave Mode, Phase = 0 .................................................................................... 372
Figure 58. SPI Slave Mode, Phase = 1 .................................................................................... 373
Figure 59. Serial Wire Debug Timing ....................................................................................... 379
Figure 60. BGA Package Drawing ............................................................................................ 381
Figure 61. Example PCB Land Pattern for BGA Package ...................................................... 382
Figure 62. Example Solder Stencil Pattern for BGA Package ................................................ 382
Figure 63. CSP Package Drawing ............................................................................................ 383
Figure 64. Reflow Soldering Diagram ...................................................................................... 384

## List of Tables

Table 1: Pin List and Function Table....26
Table 2: ARM Cortex-M4 Vector Table for Apollo SoC....37
Table 3: ARM Cortex-M4 Memory Map....38
Table 4: MCU System Memory Map....39
Table 5: MCU Peripheral Device Memory Map....40
Table 6: ITM Register Map....44
Table 7: STIM0 Register....45
Table 8: STIM0 Register Bits....45
Table 9: STIM1 Register....46
Table 10: STIM1 Register Bits....46
Table 11: STIM2 Register....46
Table 12: STIM2 Register Bits....46
Table 13: STIM3 Register....47
Table 14: STIM3 Register Bits....47
Table 15: STIM4 Register....47
Table 16: STIM4 Register Bits....47
Table 17: STIM5 Register....48
Table 18: STIM5 Register Bits....48
Table 19: STIM6 Register....48
Table 20: STIM6 Register Bits....48
Table 21: STIM7 Register....49
Table 22: STIM7 Register Bits....49
Table 23: STIM8 Register....49
Table 24: STIM8 Register Bits....49
Table 25: STIM9 Register....50
Table 26: STIM9 Register Bits....50
Table 27: STIM10 Register....50
Table 28: STIM10 Register Bits....50
Table 29: STIM11 Register....51
Table 30: STIM11 Register Bits....51
Table 31: STIM12 Register....51
Table 32: STIM12 Register Bits....51
Table 33: STIM13 Register....52
Table 34: STIM13 Register Bits....52
Table 35: STIM14 Register....52
Table 36: STIM14 Register Bits....52
Table 37: STIM15 Register....53
Table 38: STIM15 Register Bits....53
Table 39: STIM16 Register....53
Table 40: STIM16 Register Bits....53
Table 41: STIM17 Register....54
Table 42: STIM17 Register Bits....54
Table 43: STIM18 Register....54
Table 44: STIM18 Register Bits....54

Table 45: STIM19 Register ....55
Table 46: STIM19 Register Bits ....55
Table 47: STIM20 Register ....55
Table 48: STIM20 Register Bits ....55
Table 49: STIM21 Register ....56
Table 50: STIM21 Register Bits ....56
Table 51: STIM22 Register ....56
Table 52: STIM22 Register Bits ....56
Table 53: STIM23 Register ....57
Table 54: STIM23 Register Bits ....57
Table 55: STIM24 Register ....57
Table 56: STIM24 Register Bits ....57
Table 57: STIM25 Register ....58
Table 58: STIM25 Register Bits ....58
Table 59: STIM26 Register ....58
Table 60: STIM26 Register Bits ....58
Table 61: STIM27 Register ....59
Table 62: STIM27 Register Bits ....59
Table 63: STIM28 Register ....59
Table 64: STIM28 Register Bits ....59
Table 65: STIM29 Register ....60
Table 66: STIM29 Register Bits ....60
Table 67: STIM30 Register ....60
Table 68: STIM30 Register Bits ....60
Table 69: STIM31 Register ....61
Table 70: STIM31 Register Bits ....61
Table 71: TER Register ....61
Table 72: TER Register Bits ....61
Table 73: TPR Register....62
Table 74: TPR Register Bits ....62
Table 75: TCR Register ....62
Table 76: TCR Register Bits....62
Table 77: LOCKAREG Register ....63
Table 78: LOCKAREG Register Bits....63
Table 79: LOCKSREG Register....64
Table 80: LOCKSREG Register Bits ....64
Table 81: PID4 Register ....65
Table 82: PID4 Register Bits ....65
Table 83: PID5 Register ....65
Table 84: PID5 Register Bits ....65
Table 85: PID6 Register ....66
Table 86: PID6 Register Bits ....66
Table 87: PID7 Register ....66
Table 88: PID7 Register Bits ....66
Table 89: PID0 Register ....67
Table 90: PID0 Register Bits ....67

Table 91: PID1 Register ..........67
Table 92: PID1 Register Bits ..........67
Table 93: PID2 Register ..........68
Table 94: PID2 Register Bits ..........68
Table 95: PID3 Register ..........68
Table 96: PID3 Register Bits ..........68
Table 97: CID0 Register ..........69
Table 98: CID0 Register Bits ..........69
Table 99: CID1 Register ..........69
Table 100: CID1 Register Bits ..........69
Table 101: CID2 Register ..........70
Table 102: CID2 Register Bits ..........70
Table 103: CID3 Register ..........70
Table 104: CID3 Register Bits ..........70
Table 105: MCUCTRL Register Map ..........71
Table 106: CHIP_INFO Register ..........72
Table 107: CHIP_INFO Register Bits ..........72
Table 108: CHIPID0 Register ..........73
Table 109: CHIPID0 Register Bits ..........73
Table 110: CHIPID1 Register ..........73
Table 111: CHIPID1 Register Bits ..........74
Table 112: CHIPREV Register ..........74
Table 113: CHIPREV Register Bits ..........74
Table 114: SUPPLYSRC Register ..........74
Table 115: SUPPLYSRC Register Bits ..........75
Table 116: SUPPLYSTATUS Register ..........75
Table 117: SUPPLYSTATUS Register Bits ..........75
Table 118: BANDGAPEN Register ..........76
Table 119: BANDGAPEN Register Bits ..........76
Table 120: SRAMPWDINSLEEP Register ..........76
Table 121: SRAMPWDINSLEEP Register Bits ..........77
Table 122: SRAMPWRDIS Register ..........78
Table 123: SRAMPWRDIS Register Bits ..........78
Table 124: FLASHPWRDIS Register ..........79
Table 125: FLASHPWRDIS Register Bits ..........79
Table 126: ICODEFAULTADDR Register ..........80
Table 127: ICODEFAULTADDR Register Bits ..........80
Table 128: DCODEFAULTADDR Register ..........80
Table 129: DCODEFAULTADDR Register Bits ..........80
Table 130: SYSFAULTADDR Register ..........81
Table 131: SYSFAULTADDR Register Bits ..........81
Table 132: FAULTSTATUS Register ..........81
Table 133: FAULTSTATUS Register Bits ..........81
Table 134: FAULTCAPTUREEN Register ..........82
Table 135: FAULTCAPTUREEN Register Bits ..........82
Table 136: TPIUCTRL Register ..........83

Table 137: TPIUCTRL Register Bits ....83
Table 138: IOMSTR Register Map ....97
Table 139: FIFO Register ....98
Table 140: FIFO Register Bits....98
Table 141: FIFOPTR Register....98
Table 142: FIFOPTR Register Bits ....98
Table 143: TLNGTH Register....99
Table 144: TLNGTH Register Bits ....99
Table 145: FIFOTHR Register ....100
Table 146: FIFOTHR Register Bits....100
Table 147: CLKCFG Register ....100
Table 148: CLKCFG Register Bits....100
Table 149: CMD Register....101
Table 150: CMD Register Bits ....102
Table 151: CMDRPT Register ....102
Table 152: CMDRPT Register Bits ....102
Table 153: STATUS Register....103
Table 154: STATUS Register Bits ....103
Table 155: CFG Register ....104
Table 156: CFG Register Bits....104
Table 157: INTEN Register....105
Table 158: INTEN Register Bits ....105
Table 159: INTSTAT Register ....106
Table 160: INTSTAT Register Bits....106
Table 161: INTCLR Register ....107
Table 162: INTCLR Register Bits....107
Table 163: INTSET Register ....108
Table 164: INTSET Register Bits....108
Table 165: Mapping of Direct Area Access Interrupts and Corresponding REGACCINTSTAT Bits
111
Table 166: I/O Interface Interrupt Control....115
Table 167: IOSLAVE Register Map ....122
Table 168: FIFOPTR Register....123
Table 169: FIFOPTR Register Bits ....123
Table 170: FIFOCFG Register ....123
Table 171: FIFOCFG Register Bits....124
Table 172: FIFOTHR Register ....124
Table 173: FIFOTHR Register Bits....124
Table 174: FUPD Register....125
Table 175: FUPD Register Bits ....125
Table 176: FIFOCTR Register ....125
Table 177: FIFOCTR Register Bits....125
Table 178: FIFOINC Register ....126
Table 179: FIFOINC Register Bits....126
Table 180: CFG Register ....126
Table 181: CFG Register Bits....127

Table 182: PRENC Register ....127
Table 183: PRENC Register Bits....128
Table 184: IOINTCTL Register ....128
Table 185: IOINTCTL Register Bits ....128
Table 186: GENADD Register ....129
Table 187: GENADD Register Bits....129
Table 188: INTEN Register....129
Table 189: INTEN Register Bits ....129
Table 190: INTSTAT Register ....130
Table 191: INTSTAT Register Bits....130
Table 192: INTCLR Register ....131
Table 193: INTCLR Register Bits....131
Table 194: INTSET Register ....132
Table 195: INTSET Register Bits....132
Table 196: REGACCINTEN Register....133
Table 197: REGACCINTEN Register Bits ....133
Table 198: REGACCINTSTAT Register....133
Table 199: REGACCINTSTAT Register Bits....133
Table 200: REGACCINTCLR Register ....134
Table 201: REGACCINTCLR Register Bits....134
Table 202: REGACCINTSET Register....134
Table 203: REGACCINTSET Register Bits ....134
Table 204: HOST_IER Register ....135
Table 205: HOST_IER Register Bits....135
Table 206: HOST_ISR Register ....136
Table 207: HOST_ISR Register Bits....136
Table 208: HOST_WCR Register ....136
Table 209: HOST_WCR Register Bits....137
Table 210: HOST_WCS Register....137
Table 211: HOST_WCS Register Bits ....137
Table 212: FIFOCTRLO Register ....138
Table 213: FIFOCTRLO Register Bits....138
Table 214: FIFOCTRUP Register ....138
Table 215: FIFOCTRUP Register Bits....138
Table 216: FIFO Register ....139
Table 217: FIFO Register Bits....139
Table 218: Apollo Pad Function Mapping ....142
Table 219: Pad Function Color and Symbol Code ....143
Table 220: Special Pad Types....143
Table 221: I2C Pullup Resistor Selection....144
Table 222: IO Master 0 I2C Configuration ....147
Table 223:  IO Master 1 I2C Configuration ....148
Table 224: IO Master 0 4-wire SPI Configuration ....148
Table 226: IO Master 1 4-wire SPI Configuration ....149
Table 225: IO Master 0 4-wire SPI nCE Configuration ....149
Table 227: IO Master 1 4-wire SPI nCE Configuration ....150

Table 228: IO Master 0 3-wire SPI Configuration ............150
Table 229: IO Master 1 3-wire SPI Configuration ............151
Table 230: IO Slave I2C Configuration............151
Table 231: IO Slave 4-wire SPI Configuration ............151
Table 232: IO Slave 3-wire SPI Configuration ............152
Table 233: IO Master 0 I2C Loopback............152
Table 234: IO Master 1 I2C Loopback............152
Table 235: IO Master 0 4-wire SPI Loopback............153
Table 236: IO Master 1 4-wire SPI Loopback............153
Table 237: IO Master 0 3-wire SPI Loopback............153
Table 238: IO Master 1 3-wire SPI Loopback............154
Table 239: Counter/Timer Pad Configuration............155
Table 240: UART TX Configuration............156
Table 241: UART RX Configuration ............156
Table 242: UART RTS Configuration............157
Table 243: UART CTS Configuration............157
Table 244: CLKOUT Configuration............157
Table 245: ADC Analog Input Configuration ............158
Table 246: ADC Trigger Input Configuration............159
Table 247: Voltage Comparator Reference Configuration............159
Table 248: Voltage Comparator Input Configuration............159
Table 249: SWO Configuration............160
Table 250: GPIO Register Map ............161
Table 251: PADREGA Register............162
Table 252: PADREGA Register Bits............163
Table 253: PADREGB Register ............165
Table 254: PADREGB Register Bits............165
Table 255: PADREGC Register ............168
Table 256: PADREGC Register Bits............168
Table 257: PADREGD Register............170
Table 258: PADREGD Register Bits............170
Table 259: PADREGE Register ............172
Table 260: PADREGE Register Bits............173
Table 261: PADREGF Register............175
Table 262: PADREGF Register Bits ............175
Table 263: PADREGG Register............177
Table 264: PADREGG Register Bits............177
Table 265: PADREGH Register............179
Table 266: PADREGH Register Bits............179
Table 267: PADREGI Register............181
Table 268: PADREGI Register Bits ............181
Table 269: PADREGJ Register ............183
Table 270: PADREGJ Register Bits............183
Table 271: PADREGK Register............185
Table 272: PADREGK Register Bits............185
Table 273: PADREGL Register ............187

Table 274: PADREGL Register Bits ........................................................................187
Table 275: PADREGM Register ............................................................................189
Table 276: PADREGM Register Bits ......................................................................189
Table 277: CFGA Register ..................................................................................190
Table 278: CFGA Register Bits .............................................................................190
Table 279: CFGB Register ..................................................................................193
Table 280: CFGB Register Bits .............................................................................193
Table 281: CFGC Register ..................................................................................195
Table 282: CFGC Register Bits .............................................................................196
Table 283: CFGD Register ..................................................................................198
Table 284: CFGD Register Bits .............................................................................198
Table 285: CFGE Register...................................................................................201
Table 286: CFGE Register Bits .............................................................................201
Table 287: CFGF Register...................................................................................203
Table 288: CFGF Register Bits .............................................................................204
Table 289: CFGG Register ..................................................................................206
Table 290: CFGG Register Bits.............................................................................206
Table 291: PADKEY Register...............................................................................207
Table 292: PADKEY Register Bits ........................................................................207
Table 293: RDA Register ....................................................................................208
Table 294: RDA Register Bits ...............................................................................208
Table 295: RDB Register.....................................................................................208
Table 296: RDB Register Bits ...............................................................................208
Table 297: WTA Register.....................................................................................209
Table 298: WTA Register Bits ..............................................................................209
Table 299: WTB Register ....................................................................................209
Table 300: WTB Register Bits...............................................................................209
Table 301: WTSA Register...................................................................................210
Table 302: WTSA Register Bits ............................................................................210
Table 303: WTSB Register...................................................................................210
Table 304: WTSB Register Bits ............................................................................210
Table 305: WTCA Register ..................................................................................211
Table 306: WTCA Register Bits............................................................................211
Table 307: WTCB Register .................................................................................211
Table 308: WTCB Register Bits............................................................................211
Table 309: ENA Register.....................................................................................212
Table 310: ENA Register Bits ...............................................................................212
Table 311: ENB Register.....................................................................................212
Table 312: ENB Register Bits ...............................................................................212
Table 313: ENSA Register ..................................................................................213
Table 314: ENSA Register Bits .............................................................................213
Table 315: ENSB Register...................................................................................213
Table 316: ENSB Register Bits .............................................................................213
Table 317: ENCA Register ..................................................................................214
Table 318: ENCA Register Bits.............................................................................214
Table 319: ENCB Register ..................................................................................214

Table 320: ENCB Register Bits....214
Table 321: INT0EN Register....215
Table 322: INT0EN Register Bits....215
Table 323: INT0STAT Register....216
Table 324: INT0STAT Register Bits....217
Table 325: INT0CLR Register....218
Table 326: INT0CLR Register Bits....218
Table 327: INT0SET Register....220
Table 328: INT0SET Register Bits....220
Table 329: INT1EN Register....222
Table 330: INT1EN Register Bits....222
Table 331: INT1STAT Register....223
Table 332: INT1STAT Register Bits....223
Table 333: INT1CLR Register....225
Table 334: INT1CLR Register Bits....225
Table 335: INT1SET Register....226
Table 336: INT1SET Register Bits....226
Table 337: CLKGEN Register Map....233
Table 338: CALXT Register....233
Table 339: CALXT Register Bits....233
Table 340: CALRC Register....234
Table 341: CALRC Register Bits....234
Table 342: ACALCTR Register....234
Table 343: ACALCTR Register Bits....234
Table 344: OCTRL Register....235
Table 345: OCTRL Register Bits....235
Table 346: CALXT Register Bits....236
Table 347: CLKOUT Register....236
Table 348: CLKOUT Register Bits....236
Table 349: CLKKEY Register....238
Table 350: CLKKEY Register Bits....238
Table 351: CCTRL Register....238
Table 352: CCTRL Register Bits....238
Table 353: STATUS Register....239
Table 354: STATUS Register Bits....239
Table 355: HFADJ Register....240
Table 356: HFADJ Register Bits....240
Table 357: HFVAL Register....241
Table 358: HFVAL Register Bits....241
Table 359: CLOCKEN Register....241
Table 360: CLOCKEN Register Bits....241
Table 361: UARTEN Register....242
Table 362: UARTEN Register Bits....242
Table 363: INTEN Register....242
Table 364: INTEN Register Bits....242
Table 365: INTSTAT Register....243

Table 366: INTSTAT Register Bits ....243
Table 367: INTCLR Register ....244
Table 368: INTCLR Register Bits ....244
Table 369: INTSET Register ....244
Table 370: INTSET Register Bits ....244
Table 371: Alarm RPT Function ....247
Table 372: RTC Register Map ....248
Table 373: CTRLOW Register ....248
Table 374: CTRLOW Register Bits ....249
Table 375: CTRUP Register ....249
Table 376: CTRUP Register Bits ....249
Table 377: ALMLOW Register ....250
Table 378: ALMLOW Register Bits ....250
Table 379: ALMUP Register ....251
Table 380: ALMUP Register Bits ....251
Table 381: RTCCTL Register ....252
Table 382: RTCCTL Register Bits ....252
Table 383: INTEN Register ....253
Table 384: INTEN Register Bits ....253
Table 385: INTSTAT Register ....253
Table 386: INTSTAT Register Bits ....253
Table 387: INTCLR Register ....254
Table 388: INTCLR Register Bits ....254
Table 389: INTSET Register ....255
Table 390: INTSET Register Bits ....255
Table 391: CTIMER Register Map ....261
Table 392: TMR0 Register ....262
Table 393: TMR0 Register Bits ....262
Table 394: CMPRA0 Register ....262
Table 395: CMPRA0 Register Bits ....262
Table 396: CMPRB0 Register ....263
Table 397: CMPRB0 Register Bits ....263
Table 398: CTRL0 Register ....263
Table 399: CTRL0 Register Bits ....264
Table 400: TMR1 Register ....266
Table 401: TMR1 Register Bits ....266
Table 402: CMPRA1 Register ....267
Table 403: CMPRA1 Register Bits ....267
Table 404: CMPRB1 Register ....267
Table 405: CMPRB1 Register Bits ....267
Table 406: CTRL1 Register ....268
Table 407: CTRL1 Register Bits ....268
Table 408: TMR2 Register ....270
Table 409: TMR2 Register Bits ....271
Table 410: CMPRA2 Register ....271
Table 411: CMPRA2 Register Bits ....271

Table 412: CMPRB2 Register ....271
Table 413: CMPRB2 Register Bits....272
Table 414: CTRL2 Register....272
Table 415: CTRL2 Register Bits ....272
Table 416: TMR3 Register ....275
Table 417: TMR3 Register Bits....275
Table 418: CMPRA3 Register....275
Table 419: CMPRA3 Register Bits ....275
Table 420: CMPRB3 Register ....276
Table 421: CMPRB3 Register Bits....276
Table 422: CTRL3 Register....276
Table 423: CTRL3 Register Bits ....277
Table 424: INTEN Register....279
Table 425: INTEN Register Bits ....280
Table 426: INTSTAT Register ....280
Table 427: INTSTAT Register Bits....280
Table 428: INTCLR Register ....281
Table 429: INTCLR Register Bits....281
Table 430: INTSET Register ....282
Table 431: INTSET Register Bits....282
Table 432: WDT Register Map ....285
Table 433: CFG Register ....285
Table 434: CFG Register Bits....285
Table 435: RSTRT Register ....286
Table 436: RSTRT Register Bits....286
Table 437: LOCK Register ....287
Table 438: LOCK Register Bits....287
Table 439: INTEN Register....287
Table 440: INTEN Register Bits ....287
Table 441: INTSTAT Register ....288
Table 442: INTSTAT Register Bits....288
Table 443: INTCLR Register ....288
Table 444: INTCLR Register Bits....289
Table 445: INTSET Register ....289
Table 446: INTSET Register Bits....289
Table 447: RSTGEN Register Map ....292
Table 448: CFG Register ....292
Table 449: CFG Register Bits....293
Table 450: SWPOI Register ....293
Table 451: SWPOI Register Bits ....293
Table 452: SWPOR Register ....294
Table 453: SWPOR Register Bits....294
Table 454: STAT Register....294
Table 455: STAT Register Bits ....294
Table 456: CLRSTAT Register ....295
Table 457: CLRSTAT Register Bits....295

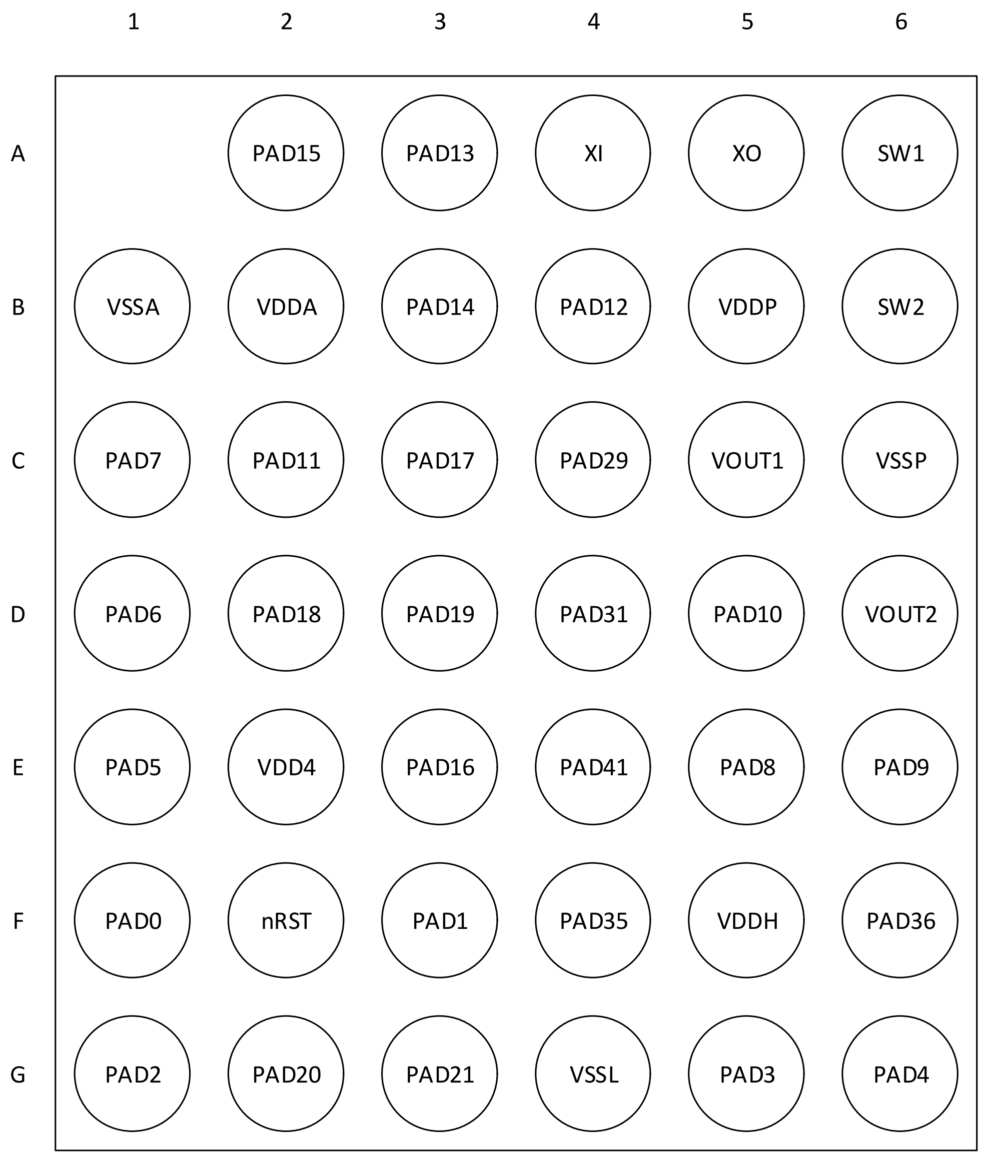
Table 458: INTEN Register ....296
Table 459: INTEN Register Bits ....296
Table 460: INTSTAT Register ....296
Table 461: INTSTAT Register Bits ....296
Table 462: INTCLR Register ....297
Table 463: INTCLR Register Bits ....297
Table 464: INTSET Register ....297
Table 465: INTSET Register Bits ....298
Table 466: UART Register Map ....301
Table 467: DR Register ....301
Table 468: DR Register Bits ....302
Table 469: RSR Register ....302
Table 470: RSR Register Bits ....302
Table 471: FR Register ....303
Table 472: FR Register Bits ....303
Table 473: ILPR Register ....304
Table 474: ILPR Register Bits ....304
Table 475: IBRD Register ....305
Table 476: IBRD Register Bits ....305
Table 477: FBRD Register ....305
Table 478: FBRD Register Bits ....305
Table 479: LCRH Register ....306
Table 480: LCRH Register Bits ....306
Table 481: CR Register ....307
Table 482: CR Register Bits ....307
Table 483: IFLS Register ....308
Table 484: IFLS Register Bits ....308
Table 485: IER Register ....309
Table 486: IER Register Bits ....309
Table 487: IES Register ....310
Table 488: IES Register Bits ....310
Table 489: MIS Register ....311
Table 490: MIS Register Bits ....311
Table 491: IEC Register ....312
Table 492: IEC Register Bits ....312
Table 493: One SLOT Configuration Register ....316
Table 494: 10.6 ADC Sample Format ....317
Table 495: Per Slot Sample Accumulator ....317
Table 496: Accumulator Scaling ....317
Table 497: FIFO Register ....318
Table 498: Window Comparator Register ....319
Table 499: ADC Register Map ....324
Table 500: CFG Register ....324
Table 501: CFG Register Bits ....325
Table 502: STAT Register ....327
Table 503: STAT Register Bits ....327

Table 504: SWT Register ....................327
Table 505: SWT Register Bits ....................327
Table 506: SL0CFG Register ....................328
Table 507: SL0CFG Register Bits ....................328
Table 508: SL1CFG Register ....................329
Table 509: SL1CFG Register Bits ....................329
Table 510: SL2CFG Register ....................331
Table 511: SL2CFG Register Bits ....................331
Table 512: SL3CFG Register ....................333
Table 513: SL3CFG Register Bits ....................333
Table 514: SL4CFG Register ....................334
Table 515: SL4CFG Register Bits ....................334
Table 516: SL5CFG Register ....................336
Table 517: SL5CFG Register Bits ....................336
Table 518: SL6CFG Register ....................338
Table 519: SL6CFG Register Bits ....................338
Table 520: SL7CFG Register ....................339
Table 521: SL7CFG Register Bits ....................339
Table 522: WLIM Register....................341
Table 523: WLIM Register Bits ....................341
Table 524: FIFO Register ....................341
Table 525: FIFO Register Bits....................342
Table 526: INTEN Register....................342
Table 527: INTEN Register Bits ....................342
Table 528: INTSTAT Register ....................343
Table 529: INTSTAT Register Bits....................343
Table 530: INTCLR Register ....................344
Table 531: INTCLR Register Bits....................344
Table 532: INTSET Register ....................345
Table 533: INTSET Register Bits....................345
Table 534: VCOMP Register Map ....................347
Table 535: CFG Register ....................348
Table 536: CFG Register Bits....................348
Table 537: STAT Register....................350
Table 538: STAT Register Bits ....................350
Table 539: PWDKEY Register....................350
Table 540: PWDKEY Register Bits ....................351
Table 541: INTEN Register....................351
Table 542: INTEN Register Bits ....................351
Table 543: INTSTAT Register ....................352
Table 544: INTSTAT Register Bits....................352
Table 545: INTCLR Register ....................352
Table 546: INTCLR Register Bits....................353
Table 547: INTSET Register ....................353
Table 548: INTSET Register Bits....................353
Table 549: Absolute Maximum Ratings....................356

Table 550: Recommended Operating Conditions....................357
Table 551: Current Consumption....................358
Table 552: Power Mode Transitions....................359
Table 553: Clocks/Oscillators....................360
Table 554: Analog to Digital Converter (ADC)....................361
Table 555: Buck Converter....................363
Table 556: Power-On Reset (POR) and Brown-Out Detector (BOD)....................364
Table 557: Resets....................365
Table 558: Voltage Comparator (VCOMP)....................366
Table 559: Internal DAC Reference for VCOMP....................367
Table 560: Inter-Integrated Circuit (I2C) Interface....................368
Table 561:  Serial Peripheral Interface (SPI) Master Interface....................369
Table 562: Serial Peripheral Interface (SPI) Slave Interface....................371
Table 563: Universal Asynchronous Receiver/Transmitter (UART)....................374
Table 564: Counter/Timer (CTIMER)....................375
Table 565: Flash Memory....................376
Table 566: General Purpose Input/Output (GPIO)....................377
Table 567: Serial Wire Debug (SWD)....................379
Table 568: Reflow Soldering Requirements (Pb-free assembly)....................384
Table 569: Ordering Information....................395
Table 570: Document Revision History....................396

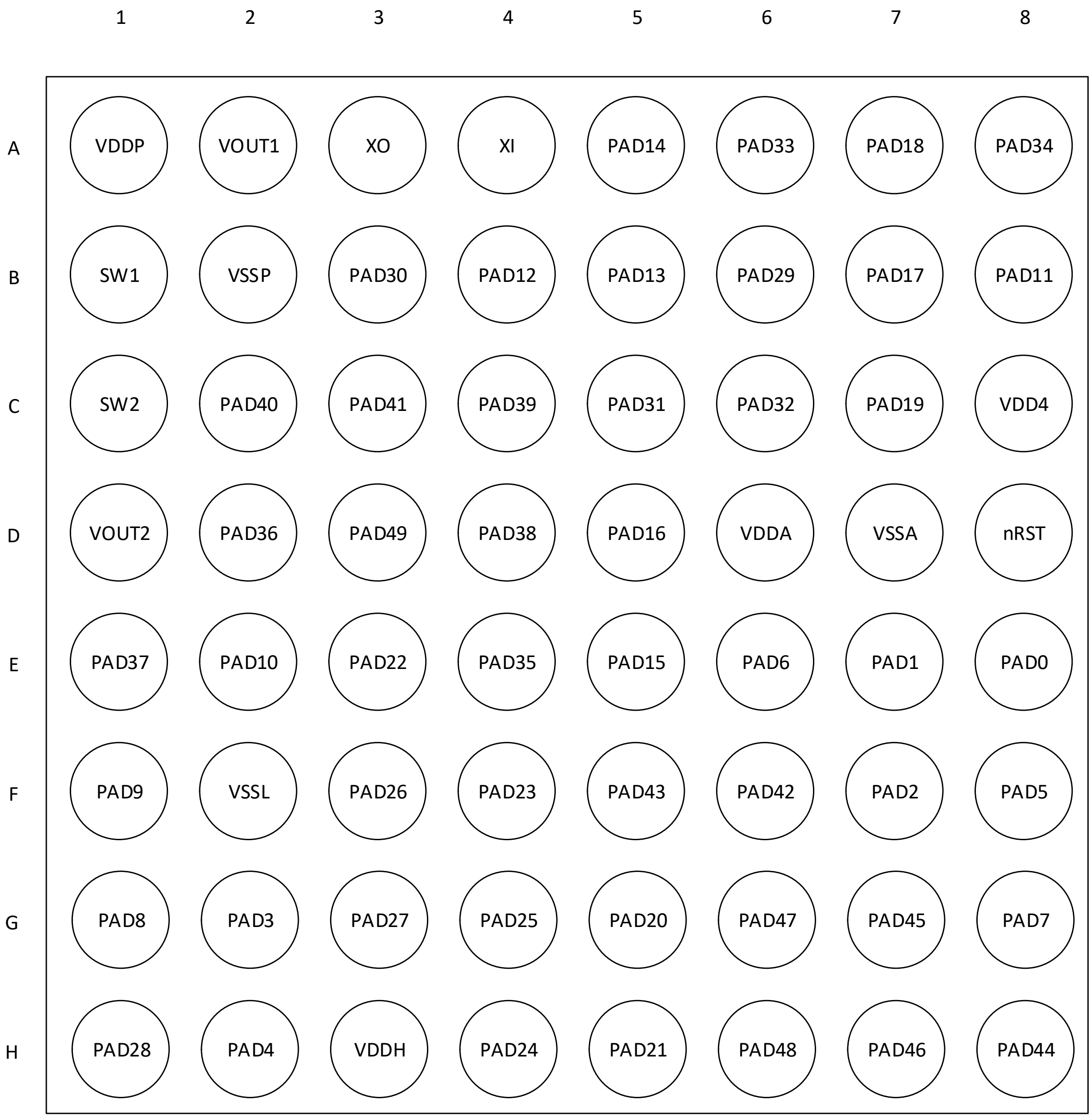
# 1. Apollo SoC Package Pins

## 1.1 Pin Configuration



|   | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| A |  | PAD15 | PAD13 | XI | XO | SW1 |
| B | VSSA | VDDA | PAD14 | PAD12 | VDDP | SW2 |
| C | PAD7 | PAD11 | PAD17 | PAD29 | VOUT1 | VSSP |
| D | PAD6 | PAD18 | PAD19 | PAD31 | PAD10 | VOUT2 |
| E | PAD5 | VDD4 | PAD16 | PAD41 | PAD8 | PAD9 |
| F | PAD0 | nRST | PAD1 | PAD35 | VDDH | PAD36 |
| G | PAD2 | PAD20 | PAD21 | VSSL | PAD3 | PAD4 |

**Figure 1. CSP Pin Configuration Diagram (Top View — Balls on Bottom)**



| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| A | VDDP | VOUT1 | XO | XI | PAD14 | PAD33 | PAD18 | PAD34 |
| B | SW1 | VSSP | PAD30 | PAD12 | PAD13 | PAD29 | PAD17 | PAD11 |
| C | SW2 | PAD40 | PAD41 | PAD39 | PAD31 | PAD32 | PAD19 | VDD4 |
| D | VOUT2 | PAD36 | PAD49 | PAD38 | PAD16 | VDDA | VSSA | nRST |
| E | PAD37 | PAD10 | PAD22 | PAD35 | PAD15 | PAD6 | PAD1 | PAD0 |
| F | PAD9 | VSSL | PAD26 | PAD23 | PAD43 | PAD42 | PAD2 | PAD5 |
| G | PAD8 | PAD3 | PAD27 | PAD25 | PAD20 | PAD47 | PAD45 | PAD7 |
| H | PAD28 | PAD4 | VDDH | PAD24 | PAD21 | PAD48 | PAD46 | PAD44 |

**Figure 2. BGA Pin Configuration Diagram (Top View — Balls on Bottom)**

## 1.2 Pin Connections

The following table lists the external pins of the  and their available functions.

**Table 1: Pin List and Function Table**

| BGA Pin Number | CSP Pin Number | GPIO Pad Number | Function Select Number | Pad Function Name | Description | Pin Type |
|---|---|---|---|---|---|---|
| POWER | | | | | | |
| A1 | B5 | - | - | VDDP | VDD Supply for Buck Converters | Power |
| B2 | C6 | - | - | VSSP | Ground Connection for I/O Pads | Ground |
| D6 | B2 | - | - | VDDA | Analog Voltage Supply | Power |
| D7 | B1 | - | - | VSSA | Ground for Analog Supply | Ground |
| C8 | E2 | - | - | VDD4 | Must be connected to VDDP | Power |
| F2 | G4 | - | - | VSSL | | Ground |
| H3 | F5 | - | - | VDDH | VDD Supply for I/O Pads | Power |
| BUCK | | | | | | |
| A2 | C5 | - | - | VOUT1 | Buck Converter Voltage Output Supply 1 | Power |
| B1 | A6 | - | - | SW1 | Buck Converter 1 Inductor Switch | Power |
| C1 | B6 | - | - | SW2 | Buck Converter 2 Inductor Switch | Power |
| D1 | D6 | - | - | VOUT2 | Buck Converter Voltage Output Supply 2 | Power |
| OSCILLATOR | | | | | | |
| A3 | A5 | - | - | XO | 32.768 kHz Crystal Output | XT |
| A4 | A4 | - | - | XI | 32.768 kHz Crystal Input | XT |
| RESET | | | | | | |
| D8 | F2 | - | - | nRST | External Reset Input | Input/Output |
| GPIO | | | | | | |
| E8 | F1 | 0 | 0 | SLSCL | I$^{2}$C Slave Clock | Input |
| | | | 1 | SLSCK | SPI Slave Clock | Input |
| | | | 2 | UARTTX | UART Transmit | Output |
| | | | 3 | GPIO0 | General Purpose I/O | Input/Output |
| | | | 4 | M0SCK | Loopback SPI Master 0 Clock | Output |
| | | | 5 | M1SCK | Loopback SPI Master 1 Clock | Output |
| | | | 6 | M0SCL | Loopback I$^{2}$C Master 0 Clock | Open Drain Output |
| | | | 7 | M1SCL | ILoopback $^{2}$C Master 1 Clock | Open Drain Output |

### Table 1: Pin List and Function Table

| BGA Pin Number | CSP Pin Number | GPIO Pad Number | Function Select Number | Pad Function Name | Description | Pin Type |
|---|---|---|---|---|---|---|
| E7 | F3 | 1 | 0 | SLSDA | I$^2$C Slave I/O Data | Bidirectional Open Drain |
| | | | 1 | SLMISO | SPI Slave Output Data | Output |
| | | | 2 | UARTRX | UART Receive | Input |
| | | | 3 | GPIO1 | General Purpose I/O | Input/Output |
| | | | 4 | M0MISO | Loopback SPI Master 0 Input Data | Input |
| | | | 5 | M1MISO | Loopback SPI Master 1 Input Data | Input |
| | | | 6 | M0SDA | Loopback I$^2$C Master 0 I/O Data | Bidirectional Open Drain |
| | | | 7 | M1SDA | Loopback I$^2$C Master 1 I/O Data | Bidirectional Open Drain |
| F7 | G1 | 2 | 0 | SLWIR3 | SPI Slave I/O Pin for 3-Wire Mode | Bidirectional 3-state |
| | | | 1 | SLMOSI | SPI Slave Input Data | Input |
| | | | 2 | CLKOUT | Oscillator Output Clock | Output |
| | | | 3 | GPIO2 | General Purpose I/O | Input/Output |
| | | | 4 | M0MOSI | Loopback SPI Master 0 Output Data | Output |
| | | | 5 | M1MOSI | Loopback SPI Master 1 Output Data | Output |
| | | | 6 | M0WIR3 | Loopback SPI Master 0 I/O Pin for 3-Wire Mode | Bidirectional 3-state |
| | | | 7 | M1WIR3 | Loopback SPI Master 1 I/O Pin for 3-Wire Mode | Bidirectional 3-state |
| G2 | G5 | 3 | 0 | TRIG0 | ADC Trigger Input 0 | Input |
| | | | 1 | SLnCE | SPI Slave Chip Enable | Input |
| | | | 2 | M1nCE4 | SPI Master 1 Chip Enable 4 | Output |
| | | | 3 | GPIO3 | General Purpose I/O | Input/Output |
| | | | 4 | M0nCE | Loopback SPI Master 0 Chip Enable | Output |
| | | | 5 | M1nCE | Loopback SPI Master 1 Chip Enable | Output |
| H2 | G6 | 4 | 0 | TRIG1 | ADC Trigger Input 1 | Input |
| | | | 1 | SLINT | Configurable Slave Interrupt | Output |
| | | | 2 | M0nCE5 | SPI Master 0 Chip Enable 5 | Output |
| | | | 3 | GPIO4 | General Purpose I/O | Input/Output |
| | | | 4 | SLINTGP | Loopback Slave Interrupt | Input |
| | | | 5 | SW0 | Buck Converter 0 Inductor Switch | Power |
| | | | 6 | CLKOUT | Oscillator Output Clock | Output |

**Table 1: Pin List and Function Table**

| BGA Pin Number | CSP Pin Number | GPIO Pad Number | Function Select Number | Pad Function Name | Description | Pin Type |
|---|---|---|---|---|---|---|
| F8 | E1 | 5 | 0 | M0SCL | $I^2C$ Master 0 Clock | Open Drain Output |
| | | | 1 | M0SCK | SPI Master 0 Clock | Output |
| | | | 2 | UARTS | UART Request To Send (RTS) | Output |
| | | | 3 | GPIO5 | General Purpose I/O | Input/Output |
| | | | 4 | M0SCK | Loopback SPI Master 0 Clock | Output |
| | | | 6 | M0SCL | Loopback $I^2C$ Master 0 Clock | Open Drain Output |
| E6 | D1 | 6 | 0 | M0SDA | $I^2C$ Master 0 Data | Bidirectional Open Drain |
| | | | 1 | M0MISO | SPI Master 0 Input Data | Input |
| | | | 2 | UACTS | UART Clear To Send (CTS) | Input |
| | | | 3 | GPIO6 | General Purpose I/O | Input/Output |
| | | | 4 | SLMISO | Loopback SPI Slave Output Data | Output |
| | | | 6 | SLSDA | Loopback $I^2C$ Slave I/O Data | Bidirectional Open Drain |
| G8 | C1 | 7 | 0 | M0WIR3 | SPI Master 0 I/O Pin for 3-Wire Mode | Bidirectional 3-state |
| | | | 1 | M0MOSI | SPI Master 0 Output Data | Output |
| | | | 2 | CLKOUT | Oscillator Output Clock | Output |
| | | | 3 | GPIO7 | General Purpose I/O | Input/Output |
| | | | 6 | SLWIR3 | Loopback SPI Slave I/O Pin for 3-Wire Mode | Bidirectional 3-state |
| G1 | E5 | 8 | 0 | M1SCL | $I^2C$ Master 1 Clock | Open Drain Output |
| | | | 1 | M1SCK | SPI Master 1 Clock | Output |
| | | | 2 | M0nCE4 | SPI Master 0 Chip Enable 4 | Output |
| | | | 3 | GPIO8 | General Purpose I/O | Input/Output |
| | | | 5 | M1SCK | Loopback SPI Master 1 Clock | Output |
| | | | 7 | M1SCL | Loopback $I^2C$ Master 1 Clock | Open Drain Output |
| F1 | E6 | 9 | 0 | M1SDA | $I^2C$ Master 1 Data | Bidirectional Open Drain |
| | | | 1 | M1MISO | SPI Master 1 Input Data | Input |
| | | | 2 | M0nCE5 | SPI Master 0 Chip Enable 5 | Output |
| | | | 3 | GPIO9 | General Purpose I/O | Input/Output |
| | | | 5 | SLMISO | Loopback SPI Slave Output Data | Output |
| | | | 7 | SLSDA | Loopback $I^2C$ Slave Data | Bidirectional Open Drain |

### Table 1: Pin List and Function Table

| BGA Pin Number | CSP Pin Number | GPIO Pad Number | Function Select Number | Pad Function Name | Description | Pin Type |
|---|---|---|---|---|---|---|
| E2 | D5 | 10 | 0 | M1WIR3 | SPI Master 1 I/O Pin for 3-Wire Mode | Bidirectional 3-state |
| | | | 1 | M1MOSI | SPI Master 1 Output Data | Output |
| | | | 2 | M0nCE6 | SPI Master 0 Chip Enable 6 | Output |
| | | | 3 | GPIO10 | General Purpose I/O | Input/Output |
| | | | 7 | SLWIR3 | Loopback SPI Slave I/O Pin for 3-Wire Mode | Bidirectional 3-state |
| B8 | C2 | 11 | 0 | RESERVED | | Input |
| | | | 1 | M0nCE0 | SPI Master 0 Chip Enable 0 | Output |
| | | | 2 | CLKOUT | Oscillator Output Clock | Output |
| | | | 3 | GPIO11 | General Purpose I/O | Input/Output |
| B4 | B4 | 12 | 0 | ADC0 | Analog to Digital Converter Input 0 | Input |
| | | | 1 | M1nCE0 | SPI Master 1 Chip Enable 0 | Output |
| | | | 2 | TCTA0 | Timer/Counter A0 | Output |
| | | | 3 | GPIO12 | General Purpose I/O | Input/Output |
| B5 | A3 | 13 | 0 | ADC1 | Analog to Digital Converter Input 1 | Input |
| | | | 1 | M1nCE1 | SPI Master 1 Chip Enable 1 | Output |
| | | | 2 | TCTB0 | Timer/Counter B0 | Output |
| | | | 3 | GPIO13 | General Purpose I/O | Input/Output |
| | | | 6 | SWO | Serial Wire Debug Output | Output |
| A5 | B3 | 14 | 0 | ADC2 | Analog to Digital Converter Input 2 | Input |
| | | | 1 | M1nCE2 | SPI Master 1 Chip Enable 2 | Output |
| | | | 2 | UARTTX | UART Transmit | Output |
| | | | 3 | GPIO14 | General Purpose I/O | Input/Output |
| E5 | A2 | 15 | 0 | ADC3 | Analog to Digital Converter Input 3 | Input |
| | | | 1 | M1nCE3 | SPI Master 1 Chip Enable 3 | Output |
| | | | 2 | UARTRX | UART Receive | Input |
| | | | 3 | GPIO15 | General Purpose I/O | Input/Output |
| D5 | E3 | 16 | 0 | ADCREF | Analog to Digital Converter Reference | Input |
| | | | 1 | M0nCE4 | SPI Master 0 Chip Enable 4 | Output |
| | | | 2 | TRIG2 | ADC Trigger Input 2 | Input |
| | | | 3 | GPIO16 | General Purpose I/O | Input/Output |
| B7 | C3 | 17 | 0 | CMPIN0 | Voltage Comparator Input 0 | Input |
| | | | 1 | M0nCE1 | SPI Master 0 Chip Enable 1 | Output |
| | | | 2 | TRIG3 | ADC Trigger Input 3 | Input |
| | | | 3 | GPIO17 | General Purpose I/O | Input/Output |

### Table 1: Pin List and Function Table

| BGA Pin Number | CSP Pin Number | GPIO Pad Number | Function Select Number | Pad Function Name | Description | Pin Type |
|---|---|---|---|---|---|---|
| A7 | D2 | 18 | 0 | CMPIN1 | Voltage Comparator Input 1 | Input |
| | | | 1 | M0nCE2 | SPI Master 0 Chip Enable 2 | Output |
| | | | 2 | TCTA1 | Timer/Counter A1 | Output |
| | | | 3 | GPIO18 | General Purpose I/O | Input/Output |
| C7 | D3 | 19 | 0 | CMPRF0 | Comparator Reference 0 | Input |
| | | | 1 | M0nCE3 | SPI Master 0 Chip Enable 3 | Output |
| | | | 2 | TCTB1 | Timer/Counter B1 | Output |
| | | | 3 | GPIO19 | General Purpose I/O | Input/Output |
| G5 | G2 | 20 | 0 | SWDCK | Software Debug Clock | Input |
| | | | 1 | M1nCE5 | SPI Master 1 Chip Enable 5 | Output |
| | | | 2 | TCTA2 | Timer/Counter A2 | Output |
| | | | 3 | GPIO20 | General Purpose I/O | Input/Output |
| H5 | G3 | 21 | 0 | SWDIO | Software Data I/O | Bidirectional 3-state |
| | | | 1 | M1nCE6 | SPI Master 1 Chip Enable 6 | Output |
| | | | 2 | TCTB2 | Timer/Counter B2 | Output |
| | | | 3 | GPIO21 | General Purpose I/O | Input/Output |
| E3 | - | 22 | 0 | UARTTX | UART Transmit | Output |
| | | | 1 | M1nCE7 | SPI Master 1 Chip Enable 7 | Output |
| | | | 2 | TCTA3 | Timer/Counter A3 | Output |
| | | | 3 | GPIO22 | General Purpose I/O | Input/Output |
| F4 | - | 23 | 0 | UARTRX | UART Receive | Input |
| | | | 1 | M0nCE0 | SPI Master 0 Chip Enable 0 | Output |
| | | | 2 | TCTB3 | Timer/Counter B3 | Output |
| | | | 3 | GPIO23 | General Purpose I/O | Input/Output |
| H4 | - | 24 | 1 | M0nCE1 | SPI Master 0 Chip Enable 1 | Output |
| | | | 2 | CLKOUT | Oscillator Output Clock | Output |
| | | | 3 | GPIO24 | General Purpose I/O | Input/Output |
| G4 | - | 25 | 1 | M0nCE2 | SPI Master 0 Chip Enable 2 | Output |
| | | | 2 | TCTA0 | Timer/Counter A0 | Output |
| | | | 3 | GPIO25 | General Purpose I/O | Input/Output |
| F3 | - | 26 | 1 | M0nCE3 | SPI Master 0 Chip Enable 3 | Output |
| | | | 2 | TCTB0 | Timer/Counter B0 | Output |
| | | | 3 | GPIO26 | General Purpose I/O | Input/Output |
| G3 | - | 27 | 1 | M1nCE4 | SPI Master 1 Chip Enable 4 | Output |
| | | | 2 | TCTA1 | Timer/Counter A1 | Output |
| | | | 3 | GPIO27 | General Purpose I/O | Input/Output |

**Table 1: Pin List and Function Table**

| BGA Pin Number | CSP Pin Number | GPIO Pad Number | Function Select Number | Pad Function Name | Description | Pin Type |
|---|---|---|---|---|---|---|
| H1 | - | 28 | 1 | M1nCE5 | SPI Master 1 Chip Enable 5 | Output |
| | | | 2 | TCTB1 | Timer/Counter B1 | Output |
| | | | 3 | GPIO28 | General Purpose I/O | Input/Output |
| B6 | C4 | 29 | 0 | ADC4 | Analog to Digital Converter Input 4 | Input |
| | | | 1 | M1nCE6 | SPI Master 1 Chip Enable 6 | Output |
| | | | 2 | TCTA2 | Timer/Counter A2 | Output |
| | | | 3 | GPIO29 | General Purpose I/O | Input/Output |
| B3 | - | 30 | 0 | ADC5 | Analog to Digital Converter Input 5 | Input |
| | | | 1 | M1nCE7 | SPI Master 1 Chip Enable 7 | Output |
| | | | 2 | TCTB2 | Timer/Counter B2 | Output |
| | | | 3 | GPIO30 | General Purpose I/O | Input/Output |
| C5 | D4 | 31 | 0 | ADC6 | Analog to Digital Converter Input 6 | Input |
| | | | 1 | M0nCE4 | SPI Master 0 Chip Enable 4 | Output |
| | | | 2 | TCTA3 | Timer/Counter A3 | Output |
| | | | 3 | GPIO31 | General Purpose I/O | Input/Output |
| C6 | - | 32 | 0 | ADC7 | Analog to Digital Converter Input 7 | Input |
| | | | 1 | M0nCE5 | SPI Master 0 Chip Enable 5 | Output |
| | | | 2 | TCTB3 | Timer/Counter B3 | Output |
| | | | 3 | GPIO32 | General Purpose I/O | Input/Output |
| A6 | - | 33 | 0 | CMPRF1 | Comparator Reference 1 | Input |
| | | | 1 | M0nCE6 | SPI Master 0 Chip Enable 6 | Output |
| | | | 3 | GPIO33 | General Purpose I/O | Input/Output |
| A8 | - | 34 | 0 | CMPRF2 | Comparator Reference 2 | Input |
| | | | 1 | M0nCE7 | SPI Master 0 Chip Enable 7 | Output |
| | | | 3 | GPIO34 | General Purpose I/O | Input/Output |
| E4 | F4 | 35 | 1 | M1nCE0 | SPI Master 1 Chip Enable 0 | Output |
| | | | 2 | UARTTX | UART Transmit | Output |
| | | | 3 | GPIO35 | General Purpose I/O | Input/Output |
| D2 | F6 | 36 | 1 | M1nCE1 | SPI Master 1 Chip Enable 1 | Output |
| | | | 2 | UARTRX | UART Receive | Input |
| | | | 3 | GPIO36 | General Purpose I/O | Input/Output |
| E1 | - | 37 | 0 | TRIG0 | ADC Trigger Input 0 | Input |
| | | | 1 | M1nCE2 | SPI Master 1 Chip Enable 2 | Output |
| | | | 2 | UARTS | UART Request To Send (RTS) | Output |
| | | | 3 | GPIO37 | General Purpose I/O | Input/Output |

### Table 1: Pin List and Function Table

| BGA Pin Number | CSP Pin Number | GPIO Pad Number | Function Select Number | Pad Function Name | Description | Pin Type |
|---|---|---|---|---|---|---|
| D4 | - | 38 | 0 | TRIG1 | ADC Trigger Input 1 | Input |
| | | | 1 | M1nCE3 | SPI Master 1 Chip Enable 3 | Output |
| | | | 2 | UACTS | UART Clear To Send (CTS) | Input |
| | | | 3 | GPIO38 | General Purpose I/O | Input/Output |
| C4 | - | 39 | 0 | TRIG2 | ADC Trigger Input 2 | Input |
| | | | 1 | UARTTX | UART Transmit | Output |
| | | | 2 | CLKOUT | Oscillator Output Clock | Output |
| | | | 3 | GPIO39 | General Purpose I/O | Input/Output |
| C2 | - | 40 | 0 | TRIG3 | ADC Trigger Input 3 | Input |
| | | | 1 | UARTRX | UART Receive | Input |
| | | | 3 | GPIO40 | General Purpose I/O | Input/Output |
| C3 | E4 | 41 | 0 | TRIG4 | ADC Trigger Input 4 | Input |
| | | | 2 | SWO | Serial Wire Debug Output | Output |
| | | | 3 | GPIO41 | General Purpose I/O | Input/Output |
| F6 | - | 42 | 0 | TRIG5 | ADC Trigger Input 5 | Input |
| | | | 1 | M0nCE0 | SPI Master 0 Chip Enable 0 | Output |
| | | | 2 | TCTA0 | Timer/Counter A0 | Output |
| | | | 3 | GPIO42 | General Purpose I/O | Input/Output |
| F5 | - | 43 | 0 | TRIG6 | ADC Trigger Input 6 | Input |
| | | | 1 | M0nCE1 | SPI Master 0 Chip Enable 1 | Output |
| | | | 2 | TCTB0 | Timer/Counter B0 | Output |
| | | | 3 | GPIO43 | General Purpose I/O | Input/Output |
| H8 | -- | 44 | 0 | TRIG7 | ADC Trigger Input 7 | Input |
| | | | 1 | M0nCE2 | SPI Master 0 Chip Enable 2 | Output |
| | | | 2 | TCTA1 | Timer/Counter A1 | Output |
| | | | 3 | GPIO44 | General Purpose I/O | Input/Output |
| G7 | - | 45 | 1 | M0nCE3 | SPI Master 0 Chip Enable 3 | Output |
| | | | 2 | TCTB1 | Timer/Counter B1 | Output |
| | | | 3 | GPIO45 | General Purpose I/O | Input/Output |
| H7 | - | 46 | 1 | M0nCE4 | SPI Master 0 Chip Enable 4 | Output |
| | | | 2 | TCTA2 | Timer/Counter A2 | Output |
| | | | 3 | GPIO46 | General Purpose I/O | Input/Output |
| G6 | - | 47 | 1 | M0nCE5 | SPI Master 0 Chip Enable 5 | Output |
| | | | 2 | TCTB2 | Timer/Counter B2 | Output |
| | | | 3 | GPIO47 | General Purpose I/O | Input/Output |

**Table 1: Pin List and Function Table**

| BGA Pin Number | CSP Pin Number | GPIO Pad Number | Function Select Number | Pad Function Name | Description | Pin Type |
|---|---|---|---|---|---|---|
| H6 | - | 48 | 1 | M0nCE6 | SPI Master 0 Chip Enable 6 | Output |
| | | | 2 | TCTA3 | Timer/Counter A3 | Output |
| | | | 3 | GPIO48 | General Purpose I/O | Input/Output |
| D3 | - | 49 | 1 | M0nCE7 | SPI Master 0 Chip Enable 7 | Output |
| | | | 2 | TCTB3 | Timer/Counter B3 | Output |
| | | | 3 | GPIO49 | General Purpose I/O | Input/Output |

## 2. System Core

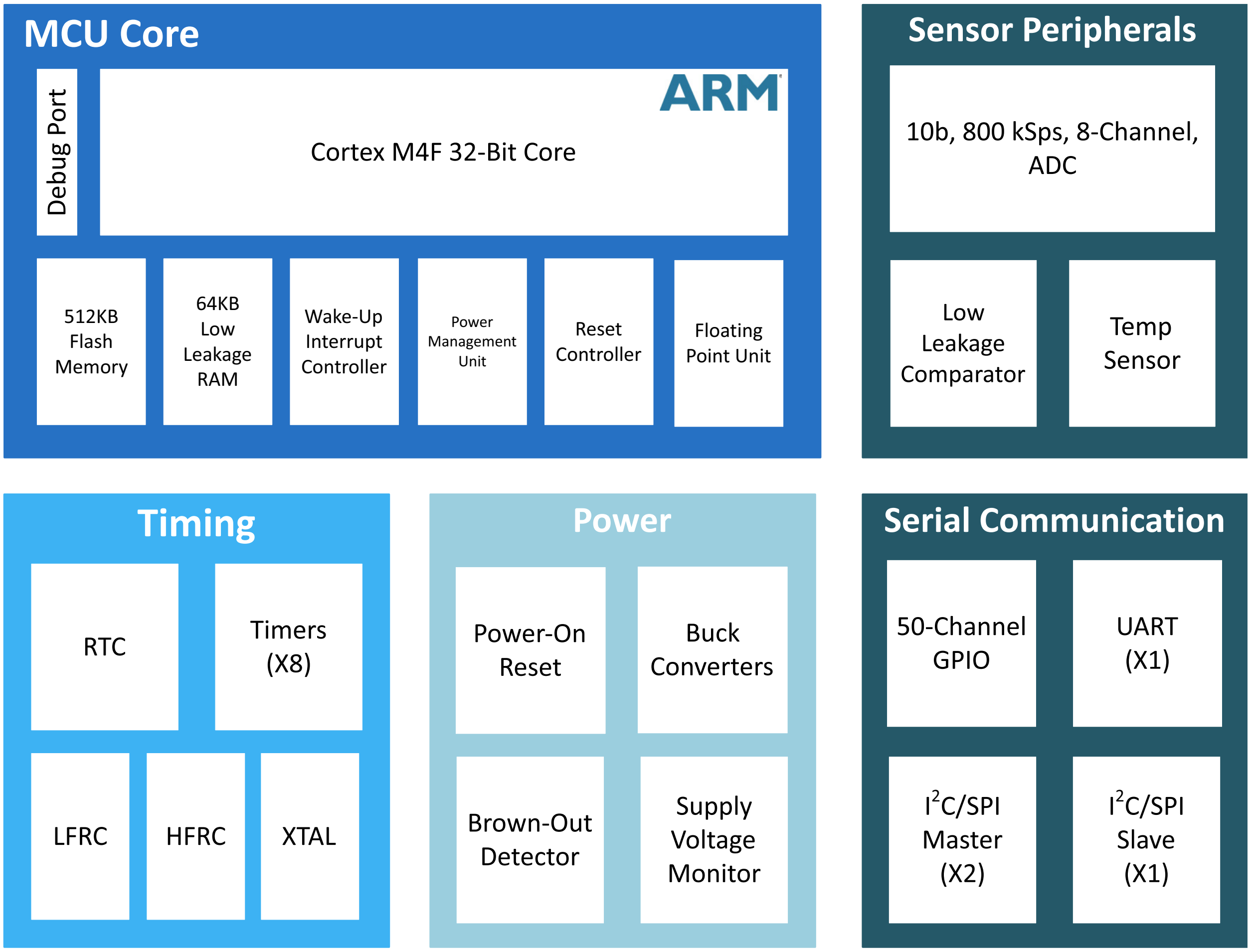


**Figure 3. Block Diagram for the Ultra-Low Power Apollo SoC**

The ultra-low power Apollo SoC, shown in Figure 3, is an ideal solution for battery-powered applications requiring sensor measurement and data analysis. In a typical system, the Apollo SoC serves as an applications processor for one or more sensors The SoC can measure analog sensor outputs using an integrated 10 bit ADC and digital sensor outputs using the integrated serial master ports. The Cortex-M4 core with Floating Point Unit (referred to throughout this document as “M4”, “M4 Core” or “Cortex-M4”) integrated in the Apollo SoC is capable of running complex data analysis and sensor fusion algorithms to process the sensor data. The Cortex-M4 core with FPU also enables accelerated time-to-market since application code may be efficiently executed in floating point form without the need to perform extensive fixed point optimizations. In other configurations, a host processor can communicate with the SoC over its serial slave port using the I$^2$C or SPI protocol.

With unprecedented energy efficiency for sensor conversion and data analysis, the Apollo SoC enables months and years of battery life for products only achieving days or months of battery life today. For example, a fitness monitoring device with days or weeks of life on a rechargeable battery could be redesigned to achieve a year or more of life on a non-rechargeable battery. Similarly the Apollo SoCs enable the use of more complex sensor processing algorithms due to its extremely low active mode power of <40 µA/MHz. By using the Apollo SoCs, the aforementioned fitness monitoring device could achieve the current multi-day or multi-week battery life while adding new computation-intensive functions like context detection and gesture recognition.

At the center of the Apollo SoC is a 32-bit ARM Cortex-M4 processor with Floating Point Unit with several tightly coupled peripherals. The Ambiq Micro implementation of the Cortex-M4 core delivers both greater performance and much lower power than 8-bit, 16-bit, and other comparable 32-bit cores. Code and data may be stored in the 512 KB Flash Memory and the 64 KB Low Leakage SRAM.

The Wake-Up Interrupt Controller (WIC) coupled with the Cortex-M4 supports sophisticated and configurable sleep state transitions with a variety of interrupt sources.

A rich set of sensor peripherals enable the monitoring of several sensors. An integrated temperature sensor enables the measurement of ambient temperature with +/-4ºC accuracy. A scalable ultra-low power Successive Approximation Register (SAR) Analog-to-Digital Converter (ADC) monitors the temperature sensor, several internal voltages, and up to eight external sensor signals. The ADC is uniquely tuned for minimum power with a configurable measurement mode that does not require MCU intervention. In addition to integrated analog sensor peripherals, an $I^2C$/SPI master port and/or UART port enables the SoC to communicate with external sensors and radios (such as Bluetooth transceivers) that have digital outputs.

The Apollo SoC also includes a set of timing peripherals based on Ambiq’s AM08XX and AM18XX Real-Time Clock (RTC) families. The RTC, timers, and counters may be driven by three different clock sources: a low frequency RC oscillator, a high frequency RC oscillator, and a 32.768 kHz crystal (XTAL) oscillator. These clock sources use the proprietary advanced calibration techniques developed for the AM08XX and AM18XX products that achieve XTAL-like accuracy with RC-like power. Additionally, the Apollo SoC includes clock reliability functions first offered in the AM08XX and AM18XX products. For example, the RTC can automatically switch from an XTAL source to an RC source in the event of an XTAL failure.

As with any ARM-based MCU, the Apollo SoC is supported by a complete suite of standard software development tools. Ambiq Micro provides drivers for all peripherals along with basic application code to shorten development times. Software debug is facilitated by the addition of an Instrumentation Trace Macrocell (ITM), a Trace Port Interface Unit (TPIU) and through the use of a Serial Wire Debugger interface (SWD).

# 3. MCU Core Details

## 3.1 Functional Overview

At the center of the Apollo SoC is a 32-bit ARM Cortex-M4 core with the floating point option. This 3-stage pipeline implementation of the ARM v7-M architecture offers highly efficient processing in a very low power design. The ARM M DAP enables debugging access via a Serial Wire Interface from outside of the MCU which allows access to all of the memory and peripheral devices of the MCU.

The M4 core offers some other advantages including:

- Single 4 GB memory architecture with all Peripherals being memory-mapped
- Low-Power Consumption Modes:
  - Active
  - Sleep
  - Deep-Sleep
  - Power-Off
- Interrupts and Events
  - NVIC – interrupt controller
  - WIC – Wake-Up Interrupt Controller
  - Sleep-on-Exit (reduces interrupt overhead, used in an ISR SW structure)
  - WFI (enter sleep modes, wait for interrupts)

The following sections provide behavioral and performance details about each of the peripherals controlled by the MCU core. Where multiple instances of a peripheral exist on Apollo (e.g., the two I$^2$C/SPI master modules), base memory addresses for the registers are provided for each and noted as INSTANCE 0, INSTANCE 1, etc.

## 3.2 Interrupts

Within the MCU, multiple peripherals can generate interrupts. In some cases, a single peripheral may be able to generate multiple different interrupts. Each interrupt signal generated by a peripheral is connected back to the M4 core in two places. First, the interrupts are connected to the Nested Vectored Interrupt Controller, NVIC, in the core. This connection provides the standard changes to program flow associated with interrupt processing. Additionally, they are connected to the WIC outside of the core, allowing the interrupt sources to wake the M4 core when it is in a deep sleep (SRPG) mode.

The MCU supports the M4 NMI as well as the normal interrupt types. For details on the Interrupt model of the M4, please see the "**Cortex-M4 Devices Generic User Guide,**" document number DUI0553A.

Below is the M4 Vector Table for Apollo SoC.

**Table 2: ARM Cortex-M4 Vector Table for Apollo SoC**

| Exception Number | IRQ Number | Offset | Vector | Peripheral/Description |
|---|---|---|---|---|
| 255 | 239 | 0x03FC | IRQ239 | |
| .<br>.<br>. | | .<br>.<br>0x00C0 | .<br>.<br>. | |
| . | | 0x00AC | IRQ27-31 | SW INT[0-4] |
| . | | 0x008C | IRQ19-26 | Stimer Compare[0:7] |
| . | | 0x0088 | IRQ18 | Stimer Capture/Overflow |
| . | | 0x0084 | IRQ17 | PDM |
| . | | 0x0080 | IRQ16 | ADC |
| . | | 0x007C | IRQ15 | UART1 |
| . | | 0x0078 | IRQ14 | UART0 |
| . | | 0x0074 | IRQ13 | Counter/Timers |
| . | | 0x0070 | IRQ12 | GPIO |
| . | | 0x006C | IRQ11 | I$^2$C/SPI Master 5 |
| . | | 0x0068 | IRQ10 | I$^2$C/SPI Master 4 |
| . | | 0x0064 | IRQ9 | I$^2$C/SPI Master 3 |
| . | | 0x0060 | IRQ8 | I$^2$C/SPI Master 2 |
| . | | 0x005C | IRQ7 | I$^2$C/SPI Master 1 |
| . | | 0x0058 | IRQ6 | I$^2$C/SPI Master 0 |
| . | | 0x0054 | IRQ5 | I$^2$C/SPI Slave Register Access |
| . | | 0x0050 | IRQ4 | I$^2$C/SPI Slave |
| . | | 0x004C | IRQ3 | Voltage Comparator |
| 18 | 2 | 0x0048 | IRQ2 | Clock Control and RTC |
| 17 | 1 | 0x0044 | IRQ1 | Watchdog Timer |
| 16 | 0 | 0x0040 | IRQ0 | Brownout Detection |
| 15 | -1 | 0x003C | Systick | |
| 14 | -2 | 0x0038 | PendSV | |
| 13 | | | Reserved | |
| 12 | | | Reserved for Debug | |
| 11 | -5 | 0x002C | SVCall | |
| 10<br>9<br>8<br>7 | | | Reserved | |
| 6 | -10 | 0x0018 | Usage Fault | |
| 5 | -11 | 0x0014 | Bus Fault | |
| 4 | -12 | 0x0010 | Memory management Fault | |
| 3 | -13 | 0x000C | Hard fault | |
| 2 | -14 | 0x0008 | NMI | Unused |
| 1 | | 0x0004 | Reset | |
| | | 0x0000 | Initial SP value | |

The Cortex-M4 allows the user to assign various interrupts to different priority levels based on the requirements of the application. In this MCU implementation, 8 different priority levels are available.

One additional feature of the M4 interrupt architecture is the ability to relocate the Vector Table to a different address. This could be useful if the application requires a different set of interrupt service routines for a particular mode of an application. The software could move the Vector Table into SRAM and reassign the interrupt service routine entry addresses as needed.

## 3.3 Memory Map

ARM has a well-defined memory map for devices based on the ARM v7-M Architecture. The M4 further refines this map in the area of the Peripheral and System address ranges. Below is the system memory map as defined by ARM:

**Table 3: ARM Cortex-M4 Memory Map**

| Address | Name | Executable | Description |
|---|---|---|---|
| 0x00000000 – 0x1FFFFFFF | Code | Y | ROM or Flash Memory |
| 0x20000000 – 0x3FFFFFFF | Reserved | N | Reserved |
| 0x40000000 – 0x5FFFFFFF | Peripheral | N | On-chip peripheral address space |
| 0x60000000 – 0x9FFFFFFF | External RAM | Y | External / Off-chip Memory |
| 0xA0000000 – 0xDFFFFFFF | External Device | N | External device memory |
| 0xE0000000 – 0xE00FFFFF | Private Peripheral Bus | N | NVIC, System timers, System Control Block |
| 0xE0100000 – 0xFFFFFFFF | Vendor_SYS | N | Vendor Defined |

The MCU-specific implementation of this memory map is as follows:

Table 4: MCU System Memory Map

| Address | Name | Executable | Description |
|---|---|---|---|
| 0x00000000 – 0x0003FFFF | Flash Bank 0 | Y | Flash Memory Instance 0 |
| 0x00040000 – 0x0007FFFF | Flash Bank 1 | Y | Flash Memory Instance 1 |
| 0x00080000 – 0x07FFFFFF | Reserved | X | No device at this address range |
| 0x08000000 – 0x08000FFF | Boot Loader ROM | Y | Execute Only Boot Loader and Flash Helper Functions. |
| 0x08001000 – 0x0FFFFFFF | Reserved | X | No device at this address range |
| 0x10000000 – 0x1000FFFF | SRAM | Y | Low-power SRAM |
| 0x10010000 – 0x3FFFFFFF | Reserved | X | No device at this address range |
| 0x40000000 – 0x4FFFFFFF | Peripheral – APB | N | APB Peripheral devices |
| 0x50000000 – 0x5FFFFFFF | Peripheral – AHB | N | AHB Peripheral devices |
| 0x60000000 – 0xDFFFFFFF | Reserved | X | No device at this address range |
| 0xE0000000 – 0xE00FFFFF | PPB | N | NVIC, System timers, System Control Block |
| 0xE0100000 – 0xEFFFFFFF | Reserved | X | No device at this address range |
| 0xF0000000 – 0xF0000FFF | Reserved | X | No device at this address range |
| 0xF0001000 – 0xFFFFFFFF | Reserved | X | No device at this address range |

Peripheral devices within the memory map are allocated on 4 KB boundaries, allowing each device up to 1024 32-bit control and status registers. Peripherals will return undefined read data when an attempt to access a register which does not exist occurs. Peripherals, whether accessed via the APB or the AHB, will always accept any write data sent to their registers without attempting to return an ERROR response. Specifically, a write to a read-only register would just become a don’t-care write.

Table 5 shows the address mapping for the peripheral devices of the Base Platform.

**Table 5: MCU Peripheral Device Memory Map**

| Address | Device |
|---|---|
| 0x40000000 – 0x400003FF | Reset / BoD Control |
| 0x40000400 – 0x40003FFF | Reserved |
| 0x40004000 – 0x400041FF | Clock Generator / RTC |
| 0x40004400 – 0x40007FFF | Reserved |
| 0x40008000 – 0x400083FF | Timers |
| 0x40008400 – 0x4000BFFF | Reserved |
| 0x4000C000 – 0x4000C3FF | Voltage Comparator |
| 0x4000C400 – 0x4000FFFF | Reserved |
| 0x40010000 – 0x400103FF | GPIO Control |
| 0x40010400 – 0x4001BFFF | Reserved |
| 0x4001C000 – 0x4001C3FF | UART |
| 0x4001C400 – 0x4001FFFF | Reserved |
| 0x40020000 – 0x400203FF | Miscellaneous Control |
| 0x40020400 – 0x40023FFF | Reserved |
| 0x40024000 – 0x400243FF | Watchdog Timer |
| 0x40024400 – 0x4FFFFFFF | Reserved |
| 0x50000000 – 0x500003FF | I$^2$C / SPI Slave |
| 0x50000400 – 0x50003FFF | Reserved |
| 0x50004000 – 0x500043FF | I$^2$C / SPI Master0 |
| 0x50004400 – 0x50004FFF | Reserved |
| 0x50005000 – 0x500053FF | I$^2$C / SPI Master1 |
| 0x50005400 – 0x50007FFF | Reserved |
| 0x50008000 – 0x500083FF | ADC |
| 0x50008400 – 0x5001FFFF | Reserved |
| 0x50020000 – 0x500203FF | Read-only Apollo Device InfoSpace |
| 0x50020400 – 0x500207FF | Flash OTP |
| 0x50020800 – 0x50020FFF | Reserved |

## 3.4 Memory Protection Unit (MPU)

The Apollo SoC includes an MPU which is a core component for memory protection. The M4 processor supports the standard ARMv7 *Protected Memory System Architecture* model. The MPU provides full support for:

- Protection regions.
- Overlapping protection regions, with ascending region priority:
  - 7 = highest priority
  - 0 = lowest priority.

- Access permissions
- Exporting memory attributes to the system.

MPU mismatches and permission violations invoke the programmable-priority MemManage fault handler. See the ARM®v7-M Architecture Reference Manual for more information.

You can use the MPU to:

- Enforce privilege rules.
- Separate processes.
- Enforce access rules.

## 3.5 System Buses

The ARM Cortex-M4 utilizes 3 instances of the AMBA AHB bus for communication with memory and peripherals. The ICode bus is designed for instruction fetches from the ‘Code’ memory space while the DCode bus is designed for data and debug accesses in that same region. The System bus is designed for fetches to the SRAM and other peripheral devices of the MCU.

The Apollo SoC maps the available SRAM memory onto an address space within the ‘Code’ memory space. This gives the user the opportunity to perform instruction and data fetches from the lower-power SRAM to effectively lower the power consumption of the MCU.

The peripherals of the Apollo SoC which are infrequently accessed are located on an AMBA APB bus. A bridge exists which translates the accesses from the System AHB to the APB. Accesses to these peripherals will inject a single wait-state on the AHB during any access cycle.

## 3.6 Power Management

The Power Management Unit (PMU) is a finite-state machine that controls the transitions of the MCU between power modes. When moving from Active Mode to Deep Sleep Mode, the PMU manages the state-retention capability of the registers within the Cortex-M4 core and also controls the shutdown of the voltage regulators of the MCU. Once in the Deep Sleep Mode, the PMU, in conjunction with the Wake-Up Interrupt Controller, waits for a wakeup event. When the event is observed, the PMU begins the power restoration process by re-enabling the on-chip voltage regulators and restoring the CPU register state. The M4 is then returned to active mode once all state is ready.

The Apollo SoC power modes are described in the subsequent discussion along with the operation of the PMU.

### *3.6.1 Cortex-M4 Power Modes*

The ARM Cortex-M4 defines the following 3 power modes:

- Active
- Sleep
- Deep Sleep

Each mode is described below.

### *3.6.1.1 Active Mode*

In the Active Mode, the M4 is powered up, clocks are active, and instructions are being executed. In this mode, the M4 expects all (enabled) devices attached to the AHB and APB to be powered and clocked for normal access. All of the non-debug ARM clocks (FCLK, HCLK) are active in this state.

To transition from the Active Mode to any of the lower-power modes, a specific sequence of instructions is executed on the M4 core. First, specific bits in the *ARMv7-M System Control Register* must be set to determine the mode to enter. See page B3-269 of the *ARMv7-M Architecture Reference Manual* for more details.

After the SCR is setup, code can enter the low-power states using one of the 3 following methods:

- Execute a Wait-For-Interrupt (WFI) instruction.
- WFE (Wait-For-Event) instruction not supported.
- Set the SLEEPONEXIT bit of the SCR such that the exit from an ISR will automatically return to a sleep state.

The M4 will enter a low-power mode after one of these are performed (assuming all conditions are met) and remain there until some event causes the core to return to Active Mode. The possible reasons to return to Active Mode are:

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

#### *3.6.1.2 Sleep Mode*

In the Sleep Mode, the M4 is powered up, but the clocks (HCLK, FCLK) are not active. The power supply is still applied to the M4 logic such that it can immediately become active on a wakeup event and begin executing instructions.

#### *3.6.1.3 Deep Sleep Mode*

In the Deep Sleep Mode, the M4 enters SRPG mode where the main power is removed, but the flops retain their state. The clocks are not active, and the MCU clock sources for HCLK and FCLK can be deactivated. To facilitate the removal of the source supply and entry into SRPG mode, the M4 will handshake with the Wake-up Interrupt Controller and Power Management Unit and set up the possible wakeup conditions.

## 3.7 Debug Interfaces

An external debugger can be connected to the MCU using the ARM Serial Wire Debug (SWD) interface. The SWD interface is a 2-wire interface that is supported by a variety of off-the-shelf commercial debuggers, enabling customers to utilize their development environment of choice.

Several useful debug facilities are provided in the Apollo SoC:

- Instruction Trace Macrocell (ITM)
- Trace Port Interface Unit (TPIU)
- Faulting Address Trapping Hardware

### *3.7.1 Instrumentation Trace Macrocell (ITM)*

For system trace the processor integrates an *Instrumentation Trace Macrocell* (ITM) alongside data watchpoints and a profiling unit. To enable simple and cost-effective profiling of the system events these generate, a *Serial Wire Viewer* (SWV) can export a stream of software-generated messages, data trace, and profiling information through a single pin.

### *3.7.2 Trace Port Interface Unit (TPIU)*

The Apollo SoC includes a Cortex-M4 Trace Port Interface Unit (TPIU). The TPIU is an ARM IP component that acts as a bridge between the on-chip trace data from the ITM and the single pin supporting the Serial Wire Viewer Protocol.

The TPIU includes a Trace Output Serializer that can format and send the SWV protocol in either a Manchester encoded form or as a standard UART stream.

### *3.7.3 Faulting Address Trapping Hardware*

The Apollo SoC offers an optional facility for trapping the address associated with bus faults occurring on any of the three AMBA AHB buses on the chip. This facility must be specifically enabled so that energy is not wasted when one is not actively debugging.

## 3.8 ITM Registers

**ARM ITM Registers**

**INSTANCE 0 BASE ADDRESS:**0x00000000

### *3.8.1 Register Memory Map*

**Table 6: ITM Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0xE0000000 | STIM0 | Stimulus Port Register 0 |
| 0xE0000004 | STIM1 | Stimulus Port Register 1 |
| 0xE0000008 | STIM2 | Stimulus Port Register 2 |
| 0xE000000C | STIM3 | Stimulus Port Register 3 |
| 0xE0000010 | STIM4 | Stimulus Port Register 4 |
| 0xE0000014 | STIM5 | Stimulus Port Register 5 |
| 0xE0000018 | STIM6 | Stimulus Port Register 6 |
| 0xE000001C | STIM7 | Stimulus Port Register 7 |
| 0xE0000020 | STIM8 | Stimulus Port Register 8 |
| 0xE0000024 | STIM9 | Stimulus Port Register 9 |
| 0xE0000028 | STIM10 | Stimulus Port Register 10 |
| 0xE000002C | STIM11 | Stimulus Port Register 11 |
| 0xE0000030 | STIM12 | Stimulus Port Register 12 |
| 0xE0000034 | STIM13 | Stimulus Port Register 13 |
| 0xE0000038 | STIM14 | Stimulus Port Register 14 |
| 0xE000003C | STIM15 | Stimulus Port Register 15 |
| 0xE0000040 | STIM16 | Stimulus Port Register 16 |
| 0xE0000044 | STIM17 | Stimulus Port Register 17 |
| 0xE0000048 | STIM18 | Stimulus Port Register 18 |
| 0xE000004C | STIM19 | Stimulus Port Register 19 |
| 0xE0000050 | STIM20 | Stimulus Port Register 20 |
| 0xE0000054 | STIM21 | Stimulus Port Register 21 |
| 0xE0000058 | STIM22 | Stimulus Port Register 22 |
| 0xE000005C | STIM23 | Stimulus Port Register 23 |
| 0xE0000060 | STIM24 | Stimulus Port Register 24 |
| 0xE0000064 | STIM25 | Stimulus Port Register 25 |
| 0xE0000068 | STIM26 | Stimulus Port Register 26 |
| 0xE000006C | STIM27 | Stimulus Port Register 27 |
| 0xE0000070 | STIM28 | Stimulus Port Register 28 |
| 0xE0000074 | STIM29 | Stimulus Port Register 29 |
| 0xE0000078 | STIM30 | Stimulus Port Register 30 |
| 0xE000007C | STIM31 | Stimulus Port Register 31 |
| 0xE0000E00 | TER | Trace Enable Register |

**Table 6: ITM Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0xE0000E40 | TPR | Trace Privilege Register |
| 0xE0000E80 | TCR | Trace Control Register |
| 0xE0000FB0 | LOCKAREG | Lock Access Register |
| 0xE0000FB4 | LOCKSREG | Lock Status Register |
| 0xE0000FD0 | PID4 | Peripheral Identification Register 4 |
| 0xE0000FD4 | PID5 | Peripheral Identification Register 5 |
| 0xE0000FD8 | PID6 | Peripheral Identification Register 6 |
| 0xE0000FDC | PID7 | Peripheral Identification Register 7 |
| 0xE0000FE0 | PID0 | Peripheral Identification Register 0 |
| 0xE0000FE4 | PID1 | Peripheral Identification Register 1 |
| 0xE0000FE8 | PID2 | Peripheral Identification Register 2 |
| 0xE0000FEC | PID3 | Peripheral Identification Register 3 |
| 0xE0000FF0 | CID0 | Component Identification Register 1 |
| 0xE0000FF4 | CID1 | Component Identification Register 1 |
| 0xE0000FF8 | CID2 | Component Identification Register 2 |
| 0xE0000FFC | CID3 | Component Identification Register 3 |

## 3.8.2 *ITM Registers*

### 3.8.2.1 *STIM0 Register*

**Stimulus Port Register 0**

**OFFSET:** 0xE0000000

**INSTANCE 0 ADDRESS:** 0xE0000000

Stimulus Port Register 0

**Table 7: STIM0 Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="32">STIM0</td></tr>
</table>

**Table 8: STIM0 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM0 | 0x0 | RW | Stimulus Port Register 0. |

### *3.8.2.2 STIM1 Register*

**Stimulus Port Register 1**

**OFFSET:** 0xE0000004

**INSTANCE 0 ADDRESS:** 0xE0000004

Stimulus Port Register 1

**Table 9: STIM1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 10: STIM1 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM1 | 0x0 | RW | Stimulus Port Register 1. |

### *3.8.2.3 STIM2 Register*

**Stimulus Port Register 2**

**OFFSET:** 0xE0000008

**INSTANCE 0 ADDRESS:** 0xE0000008

Stimulus Port Register 2

**Table 11: STIM2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 12: STIM2 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM2 | 0x0 | RW | Stimulus Port Register 2. |

### *3.8.2.4 STIM3 Register*

**Stimulus Port Register 3**

**OFFSET:** 0xE000000C

**INSTANCE 0 ADDRESS:** 0xE000000C

Stimulus Port Register 3

**Table 13: STIM3 Register**

| 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STIM3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 14: STIM3 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM3 | 0x0 | RW | Stimulus Port Register 3. |

### *3.8.2.5 STIM4 Register*

**Stimulus Port Register 4**

**OFFSET:** 0xE0000010

**INSTANCE 0 ADDRESS:** 0xE0000010

Stimulus Port Register 4

**Table 15: STIM4 Register**

| 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STIM4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 16: STIM4 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM4 | 0x0 | RW | Stimulus Port Register 4. |

#### *3.8.2.6 STIM5 Register*

**Stimulus Port Register 5**

**OFFSET:** 0xE0000014

**INSTANCE 0 ADDRESS:** 0xE0000014

Stimulus Port Register 5

**Table 17: STIM5 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 18: STIM5 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM5 | 0x0 | RW | Stimulus Port Register 5. |

#### *3.8.2.7 STIM6 Register*

**Stimulus Port Register 6**

**OFFSET:** 0xE0000018

**INSTANCE 0 ADDRESS:** 0xE0000018

Stimulus Port Register 6

**Table 19: STIM6 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 20: STIM6 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM6 | 0x0 | RW | Stimulus Port Register 6. |

#### *3.8.2.8 STIM7 Register*

**Stimulus Port Register 7**

**OFFSET:** 0xE000001C

**INSTANCE 0 ADDRESS:** 0xE000001C

Stimulus Port Register 7

**Table 21: STIM7 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 22: STIM7 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM7 | 0x0 | RW | Stimulus Port Register 7. |

#### *3.8.2.9 STIM8 Register*

**Stimulus Port Register 8**

**OFFSET:** 0xE0000020

**INSTANCE 0 ADDRESS:** 0xE0000020

Stimulus Port Register 8

**Table 23: STIM8 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 24: STIM8 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM8 | 0x0 | RW | Stimulus Port Register 8. |

### 3.8.2.10 STIM9 Register

**Stimulus Port Register 9**

**OFFSET:** 0xE0000024

**INSTANCE 0 ADDRESS:** 0xE0000024

Stimulus Port Register 9

**Table 25: STIM9 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM9 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 26: STIM9 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM9 | 0x0 | RW | Stimulus Port Register 9. |

### 3.8.2.11 STIM10 Register

**Stimulus Port Register 10**

**OFFSET:** 0xE0000028

**INSTANCE 0 ADDRESS:** 0xE0000028

Stimulus Port Register 10

**Table 27: STIM10 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM10 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 28: STIM10 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM10 | 0x0 | RW | Stimulus Port Register 10. |

### *3.8.2.12 STIM11 Register*

**Stimulus Port Register 11**

**OFFSET:** 0xE000002C

**INSTANCE 0 ADDRESS:** 0xE000002C

Stimulus Port Register 11

**Table 29: STIM11 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM11 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 30: STIM11 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM11 | 0x0 | RW | Stimulus Port Register 11. |

### *3.8.2.13 STIM12 Register*

**Stimulus Port Register 12**

**OFFSET:** 0xE0000030

**INSTANCE 0 ADDRESS:** 0xE0000030

Stimulus Port Register 12

**Table 31: STIM12 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM12 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 32: STIM12 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM12 | 0x0 | RW | Stimulus Port Register 12. |

### 3.8.2.14 *STIM13 Register*

**Stimulus Port Register 13**

**OFFSET:** 0xE0000034

**INSTANCE 0 ADDRESS:** 0xE0000034

Stimulus Port Register 13

**Table 33: STIM13 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM13 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 34: STIM13 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM13 | 0x0 | RW | Stimulus Port Register 13. |

### 3.8.2.15 *STIM14 Register*

**Stimulus Port Register 14**

**OFFSET:** 0xE0000038

**INSTANCE 0 ADDRESS:** 0xE0000038

Stimulus Port Register 14

**Table 35: STIM14 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM14 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 36: STIM14 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM14 | 0x0 | RW | Stimulus Port Register 14. |

### 3.8.2.16 *STIM15 Register*

**Stimulus Port Register 15**

**OFFSET:** 0xE000003C

**INSTANCE 0 ADDRESS:** 0xE000003C

Stimulus Port Register 15

**Table 37: STIM15 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM15 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 38: STIM15 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM15 | 0x0 | RW | Stimulus Port Register 15. |

### 3.8.2.17 *STIM16 Register*

**Stimulus Port Register 16**

**OFFSET:** 0xE0000040

**INSTANCE 0 ADDRESS:** 0xE0000040

Stimulus Port Register 16

**Table 39: STIM16 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM16 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 40: STIM16 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM16 | 0x0 | RW | Stimulus Port Register 16. |

### *3.8.2.18 STIM17 Register*

**Stimulus Port Register 17**

**OFFSET:** 0xE0000044

**INSTANCE 0 ADDRESS:** 0xE0000044

Stimulus Port Register 17

**Table 41: STIM17 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM17 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 42: STIM17 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM17 | 0x0 | RW | Stimulus Port Register 17. |

### *3.8.2.19 STIM18 Register*

**Stimulus Port Register 18**

**OFFSET:** 0xE0000048

**INSTANCE 0 ADDRESS:** 0xE0000048

Stimulus Port Register 18

**Table 43: STIM18 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM18 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 44: STIM18 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM18 | 0x0 | RW | Stimulus Port Register 18. |

### *3.8.2.20 STIM19 Register*

**Stimulus Port Register 19**

**OFFSET:** 0xE000004C

**INSTANCE 0 ADDRESS:** 0xE000004C

Stimulus Port Register 19

**Table 45: STIM19 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM19 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 46: STIM19 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM19 | 0x0 | RW | Stimulus Port Register 19. |

### *3.8.2.21 STIM20 Register*

**Stimulus Port Register 20**

**OFFSET:** 0xE0000050

**INSTANCE 0 ADDRESS:** 0xE0000050

Stimulus Port Register 20

**Table 47: STIM20 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM20 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 48: STIM20 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM20 | 0x0 | RW | Stimulus Port Register 20. |

### 3.8.2.22 *STIM21 Register*

**Stimulus Port Register 21**

**OFFSET:** 0xE0000054

**INSTANCE 0 ADDRESS:** 0xE0000054

Stimulus Port Register 21

**Table 49: STIM21 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM21 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 50: STIM21 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM21 | 0x0 | RW | Stimulus Port Register 21. |

### 3.8.2.23 *STIM22 Register*

**Stimulus Port Register 22**

**OFFSET:** 0xE0000058

**INSTANCE 0 ADDRESS:** 0xE0000058

Stimulus Port Register 22

**Table 51: STIM22 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM22 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 52: STIM22 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM22 | 0x0 | RW | Stimulus Port Register 22. |

### 3.8.2.24 *STIM23 Register*

**Stimulus Port Register 23**

**OFFSET:** 0xE000005C

**INSTANCE 0 ADDRESS:** 0xE000005C

Stimulus Port Register 23

**Table 53: STIM23 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM23 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 54: STIM23 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM23 | 0x0 | RW | Stimulus Port Register 23. |

### 3.8.2.25 *STIM24 Register*

**Stimulus Port Register 24**

**OFFSET:** 0xE0000060

**INSTANCE 0 ADDRESS:** 0xE0000060

Stimulus Port Register 24

**Table 55: STIM24 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM24 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 56: STIM24 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM24 | 0x0 | RW | Stimulus Port Register 24. |

### *3.8.2.26 STIM25 Register*

**Stimulus Port Register 25**

**OFFSET:** 0xE0000064

**INSTANCE 0 ADDRESS:** 0xE0000064

Stimulus Port Register 25

**Table 57: STIM25 Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="32">STIM25</td></tr>
</table>

**Table 58: STIM25 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM25 | 0x0 | RW | Stimulus Port Register 25. |

### *3.8.2.27 STIM26 Register*

**Stimulus Port Register 26**

**OFFSET:** 0xE0000068

**INSTANCE 0 ADDRESS:** 0xE0000068

Stimulus Port Register 26

**Table 59: STIM26 Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="32">STIM26</td></tr>
</table>

**Table 60: STIM26 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM26 | 0x0 | RW | Stimulus Port Register 26. |

### *3.8.2.28 STIM27 Register*

**Stimulus Port Register 27**

**OFFSET:** 0xE000006C

**INSTANCE 0 ADDRESS:** 0xE000006C

Stimulus Port Register 27

**Table 61: STIM27 Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="32">STIM27</td></tr>
</table>

**Table 62: STIM27 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM27 | 0x0 | RW | Stimulus Port Register 27. |

### *3.8.2.29 STIM28 Register*

**Stimulus Port Register 28**

**OFFSET:** 0xE0000070

**INSTANCE 0 ADDRESS:** 0xE0000070

Stimulus Port Register 28

**Table 63: STIM28 Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="32">STIM28</td></tr>
</table>

**Table 64: STIM28 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM28 | 0x0 | RW | Stimulus Port Register 28. |

### 3.8.2.30 *STIM29 Register*

**Stimulus Port Register 29**

**OFFSET:** 0xE0000074

**INSTANCE 0 ADDRESS:** 0xE0000074

Stimulus Port Register 29

**Table 65: STIM29 Register**

| 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STIM29 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 66: STIM29 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM29 | 0x0 | RW | Stimulus Port Register 29. |

### 3.8.2.31 *STIM30 Register*

**Stimulus Port Register 30**

**OFFSET:** 0xE0000078

**INSTANCE 0 ADDRESS:** 0xE0000078

Stimulus Port Register 30

**Table 67: STIM30 Register**

| 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| STIM30 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 68: STIM30 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM30 | 0x0 | RW | Stimulus Port Register 30. |

### 3.8.2.32 *STIM31 Register*

**Stimulus Port Register 31**

**OFFSET:** 0xE000007C

**INSTANCE 0 ADDRESS:** 0xE000007C

Stimulus Port Register 31

**Table 69: STIM31 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIM31 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 70: STIM31 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIM31 | 0x0 | RW | Stimulus Port Register 31. |

### 3.8.2.33 *TER Register*

**Trace Enable Register.**

**OFFSET:** 0xE0000E00

**INSTANCE 0 ADDRESS:** 0xE0000E00

Trace Enable Register

**Table 71: TER Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STIMENA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 72: TER Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | STIMENA | 0x0 | RW | Bit mask to enable tracing on ITM stimulus ports.<br>One bit per stimulus port. |

### 3.8.2.34 TPR Register

**Trace Privilege Register.**

**OFFSET:** 0xE0000E40

**INSTANCE 0 ADDRESS:** 0xE0000E40

Trace Privilege Register

**Table 73: TPR Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="28">RSVD</td><td colspan="4">PRIVMASK</td></tr>
</table>

**Table 74: TPR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:4 | RSVD | 0x0 | RO | RESERVED. |
| 3:0 | PRIVMASK | 0x0 | RW | Bit mask to enable tracing on ITM stimulus ports. bit[0] = stimulus ports[7:0], bit[1] = stimulus ports[15:8], bit[2] = stimulus ports[23:16], bit[3] = stimulus ports[31:24]. |

### 3.8.2.35 TCR Register

**Trace Control Register.**

**OFFSET:** 0xE0000E80

**INSTANCE 0 ADDRESS:** 0xE0000E80

Trace Control Register

**Table 75: TCR Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="8">RSVD</td><td>BUSY</td><td colspan="7">ATB_ID</td><td colspan="4">RSVD</td><td colspan="2">TS_FREQ</td><td colspan="2">TS_PRESCALE</td><td colspan="3">RSVD</td><td>SWV_ENABLE</td><td>DWT_ENABLE</td><td>SYNC_ENABLE</td><td>TS_ENABLE</td><td>ITM_ENABLE</td></tr>
</table>

**Table 76: TCR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:24 | RSVD | 0x0 | RO | RESERVED. |
| 23 | BUSY | 0x0 | RW | Set when ITM events present and being drained. |

**Table 76: TCR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 22:16 | ATB_ID | 0x0 | RW | ATB ID for CoreSight system. |
| 15:12 | RSVD | 0x0 | RO | RESERVED. |
| 11:10 | TS_FREQ | 0x0 | RW | Global Timestamp Frequency. |
| 9:8 | TS_PRESCALE | 0x0 | RW | Timestamp prescaler: 0b00 = no prescaling 0b01 = divide by 4 0b10 = divide by 16 0b11 = divide by 64. |
| 7:5 | RSVD | 0x0 | RO | RESERVED. |
| 4 | SWV_ENABLE | 0x0 | RW | Enable SWV behavior – count on TPIUEMIT and TPIUBAUD. |
| 3 | DWT_ENABLE | 0x0 | RW | Enables the DWT stimulus. |
| 2 | SYNC_ENABLE | 0x0 | RW | Enables sync packets for TPIU. |
| 1 | TS_ENABLE | 0x0 | RW | Enables differential timestamps. Differential timestamps are emitted when a packet is written to the FIFO with a non-zero timestamp counter, and when the timestamp counter overflows. Timestamps are emitted during idle times after a fixed number of cycles. This provides a time reference for packets and inter-packet gaps. |
| 0 | ITM_ENABLE | 0x0 | RW | Enable ITM. This is the master enable, and must be set before ITM Stimulus and Trace Enable registers can be written. |

### *3.8.2.36 LOCKAREG Register*

**Lock Access Register**

**OFFSET:** 0xE0000FB0

**INSTANCE 0 ADDRESS:** 0xE0000FB0

Lock Access Register

**Table 77: LOCKAREG Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="32">LOCKAREG</td></tr>
</table>

**Table 78: LOCKAREG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | LOCKAREG | 0x0 | RW | Key register value.<br>Key = 0xC5ACCE55 - Key |

### 3.8.2.37 LOCKSREG Register

**Lock Status Register**

**OFFSET:** 0xE0000FB4

**INSTANCE 0 ADDRESS:** 0xE0000FB4

Lock Status Register

**Table 79: LOCKSREG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | BYTEACC | ACCESS | PRESENT |

**Table 80: LOCKSREG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:3 | RSVD | 0x0 | RO | RESERVED. |
| 2 | BYTEACC | 0x0 | RO | You cannot implement 8-bit lock accesses. |
| 1 | ACCESS | 0x0 | RO | Write access to component is blocked. All writes are ignored, reads are permitted. |
| 0 | PRESENT | 0x1 | RO | Indicates that a lock mechanism exists for this component. |

### 3.8.2.38 *PID4 Register*

**Peripheral Identification Register 4**

**OFFSET:** 0xE0000FD0

**INSTANCE 0 ADDRESS:** 0xE0000FD0

Peripheral Identification Register 4

**Table 81: PID4 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PID4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 82: PID4 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | PID4 | 0x4 | R0 | Peripheral Identification 4. |

### 3.8.2.39 *PID5 Register*

**Peripheral Identification Register 5**

**OFFSET:** 0xE0000FD4

**INSTANCE 0 ADDRESS:** 0xE0000FD4

Peripheral Identification Register 5

**Table 83: PID5 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PID5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 84: PID5 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | PID5 | 0x0 | R0 | Peripheral Identification 5. |

### 3.8.2.40 PID6 Register

**Peripheral Identification Register 6**

**OFFSET:** 0xE0000FD8

**INSTANCE 0 ADDRESS:** 0xE0000FD8

Peripheral Identification Register 6

**Table 85: PID6 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PID6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 86: PID6 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | PID6 | 0x0 | R0 | Peripheral Identification 6. |

### 3.8.2.41 PID7 Register

**Peripheral Identification Register 7**

**OFFSET:** 0xE0000FDC

**INSTANCE 0 ADDRESS:** 0xE0000FDC

Peripheral Identification Register 7

**Table 87: PID7 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PID7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 88: PID7 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | PID7 | 0x0 | R0 | Peripheral Identification 7. |

### *3.8.2.42 PID0 Register*

**Peripheral Identification Register 0**

**OFFSET:** 0xE0000FE0

**INSTANCE 0 ADDRESS:** 0xE0000FE0

Peripheral Identification Register 0

**Table 89: PID0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PID0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 90: PID0 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | PID0 | 0x1 | R0 | Peripheral Identification 0. |

### *3.8.2.43 PID1 Register*

**Peripheral Identification Register 1**

**OFFSET:** 0xE0000FE4

**INSTANCE 0 ADDRESS:** 0xE0000FE4

Peripheral Identification Register 1

**Table 91: PID1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PID1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 92: PID1 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | PID1 | 0xb0 | R0 | Peripheral Identification 1. |

### *3.8.2.44 PID2 Register*

**Peripheral Identification Register 2**

**OFFSET:** 0xE0000FE8

**INSTANCE 0 ADDRESS:** 0xE0000FE8

Peripheral Identification Register 2

**Table 93: PID2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PID2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 94: PID2 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | PID2 | 0x3b | R0 | Peripheral Identification 2. |

### *3.8.2.45 PID3 Register*

**Peripheral Identification Register 3**

**OFFSET:** 0xE0000FEC

**INSTANCE 0 ADDRESS:** 0xE0000FEC

Peripheral Identification Register 3

**Table 95: PID3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PID3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 96: PID3 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | PID3 | 0x0 | R0 | Peripheral Identification 3. |

### *3.8.2.46 CID0 Register*

**Component Identification Register 1**

**OFFSET:** 0xE0000FF0

**INSTANCE 0 ADDRESS:** 0xE0000FF0

Component Identification Register 1

**Table 97: CID0 Register**

| 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CID0 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 98: CID0 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | CID0 | 0xd | R0 | Component Identification 1. |

### *3.8.2.47 CID1 Register*

**Component Identification Register 1**

**OFFSET:** 0xE0000FF4

**INSTANCE 0 ADDRESS:** 0xE0000FF4

Component Identification Register 1

**Table 99: CID1 Register**

| 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CID1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 100: CID1 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | CID1 | 0xe0 | R0 | Component Identification 1. |

### 3.8.2.48 CID2 Register

**Component Identification Register 2**

**OFFSET:** 0xE0000FF8

**INSTANCE 0 ADDRESS:** 0xE0000FF8

Component Identification Register 2

**Table 101: CID2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CID2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 102: CID2 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | CID2 | 0x5 | R0 | Component Identification 2. |

### 3.8.2.49 CID3 Register

**Component Identification Register 3**

**OFFSET:** 0xE0000FFC

**INSTANCE 0 ADDRESS:** 0xE0000FFC

Component Identification Register 3

**Table 103: CID3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CID3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 104: CID3 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | CID3 | 0xb1 | R0 | Component Identification 3. |

## 3.9 MCUCTRL Registers

**MCU Miscellaneous Control Logic**

**INSTANCE 0 BASE ADDRESS:**0x40020000

### *3.9.1 Register Memory Map*

**Table 105: MCUCTRL Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0x40020000 | CHIP_INFO | Chip Information Register |
| 0x40020004 | CHIPID0 | Unique Chip ID 0 |
| 0x40020008 | CHIPID1 | Unique Chip ID 1 |
| 0x4002000C | CHIPREV | Chip Revision |
| 0x40020010 | SUPPLYSRC | Memory and Core Voltage Supply Source Select Register |
| 0x40020014 | SUPPLYSTATUS | Memory and Core Voltage Supply Source Status Register |
| 0x400200FC | BANDGAPEN | Band Gap Enable |
| 0x40020140 | SRAMPWDINSLEEP | Powerdown an SRAM Bank in Deep Sleep mode |
| 0x40020144 | SRAMPWRDIS | Disables individual banks of the SRAM array |
| 0x40020148 | FLASHPWRDIS | Disables individual banks of the Flash array |
| 0x400201C0 | ICODEFAULTADDR | ICODE bus address which was present when a bus fault occurred. |
| 0x400201C4 | DCODEFAULTADDR | DCODE bus address which was present when a bus fault occurred. |
| 0x400201C8 | SYSFAULTADDR | System bus address which was present when a bus fault occurred. |
| 0x400201CC | FAULTSTATUS | Reflects the status of the bus decoders' fault detection. Any write to this register will clear all of the status bits within the register. |
| 0x400201D0 | FAULTCAPTUREEN | Enable the fault capture registers |
| 0x40020250 | TPIUCTRL | TPIU Control Register. Determines the clock enable and frequency for the M4's TPIU interface. |

### 3.9.2 MCUCTRL Registers

### 3.9.2.1 CHIP_INFO Register

**Chip Information Register**

**OFFSET:** 0x00000000

**INSTANCE 0 ADDRESS:** 0x40020000

Chip Information Register

**Table 106: CHIP_INFO Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CLASS | | | | | | | | FLASH | | | | RAM | | | | MAJORREV | | | | MINORREV | | | | PKG | | PINS | | | TEMP | | QUAL |

**Table 107: CHIP_INFO Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:24 | CLASS | 0x1 | RO | Device class.<br><br>APOLLO = 0x1 - APOLLO |
| 23:20 | FLASH | 0x4 | RO | Device flash size.<br><br>256K = 0x3 - 256K of available flash.<br>512K = 0x4 - 512K of available flash. |
| 19:16 | RAM | 0x1 | RO | Device RAM size.<br><br>32K = 0x0 - 32K of available SRAM.<br>64K = 0x1 - 64K of available SRAM. |
| 15:12 | MAJORREV | 0x1 | RO | Major device revision number. |
| 11:8 | MINORREV | 0x1 | RO | Minor device revision number. |
| 7:6 | PKG | 0x1 | RO | Device package type.<br><br>BGA = 0x2 - Ball grid array.<br>CSP = 0x3 - Chip-scale package. |
| 5:3 | PINS | 0x1 | RO | Number of pins.<br><br>41PINS = 0x1 - 41 package pins total.<br>64PINS = 0x1 - 64 package pins total. |
| 2:1 | TEMP | 0x1 | RO | Device temperature range.<br><br>COMMERCIAL = 0x0 - Commercial temperature range. |

**Table 107: CHIP_INFO Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 0 | QUAL | 0x1 | RO | Device qualified.<br><br>PROTOTYPE = 0x0 - Prototype device.<br>QUALIFIED = 0x1 - Fully qualified device. |

### 3.9.2.2 *CHIPID0 Register*

**Unique Chip ID 0**

**OFFSET:** 0x00000004

**INSTANCE 0 ADDRESS:** 0x40020004

Unique Chip ID 0

**Table 108: CHIPID0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VALUE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 109: CHIPID0 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | VALUE | 0x0 | RO | Unique chip ID 0.<br><br>APOLLO = 0x0 - Apollo CHIPID0. |

### 3.9.2.3 *CHIPID1 Register*

**Unique Chip ID 1**

**OFFSET:** 0x00000008

**INSTANCE 0 ADDRESS:** 0x40020008

Unique Chip ID 1

**Table 110: CHIPID1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VALUE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 111: CHIPID1 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | VALUE | 0x0 | RO | Unique chip ID 1.<br>APOLLO = 0x0 - Apollo CHIPID1. |

### 3.9.2.4 *CHIPREV Register*

**Chip Revision**

**OFFSET:** 0x0000000C

**INSTANCE 0 ADDRESS:** 0x4002000C

Chip Revision

**Table 112: CHIPREV Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 113: CHIPREV Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED. |

### 3.9.2.5 *SUPPLYSRC Register*

**Memory and Core Voltage Supply Source Select Register**

**OFFSET:** 0x00000010

**INSTANCE 0 ADDRESS:** 0x40020010

Memory and Core Voltage Supply Source Select Register

**Table 114: SUPPLYSRC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | COREBUCKEN | MEMBUCKEN |

**Table 115: SUPPLYSRC Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:2 | RSVD | 0x0 | RO | RESERVED. |
| 1 | COREBUCKEN | 0x0 | RW | Enables and Selects the Core Buck as the supply for the low-voltage power domain.<br>EN = 0x1 - Enable the Core Buck for the low-voltage power domain. |
| 0 | MEMBUCKEN | 0x0 | RW | Enables and select the Memory Buck as the supply for the Flash and SRAM power domain.<br>EN = 0x1 - Enable the Memory Buck as the supply for flash and SRAM. |

### 3.9.2.6 *SUPPLYSTATUS Register*

**Memory and Core Voltage Supply Source Status Register**

**OFFSET:** 0x00000014

**INSTANCE 0 ADDRESS:** 0x40020014

Memory and Core Voltage Supply Source Status Register

**Table 116: SUPPLYSTATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | COREBUCKON | MEMBUCKON |

**Table 117: SUPPLYSTATUS Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:2 | RSVD | 0x0 | RO | RESERVED. |
| 1 | COREBUCKON | 0x0 | RO | Indicates whether the Core low-voltage domain is supplied from the LDO or the Buck.<br>LDO = 0x0 - Indicates the the LDO is supplying the Core low-voltage.<br>BUCK = 0x1 - Indicates the the Buck is supplying the Core low-voltage. |
| 0 | MEMBUCKON | 0x0 | RO | Indicate whether the Memory power domain is supplied from the LDO or the Buck.<br>LDO = 0x0 - Indicates the LDO is supplying the memory power domain.<br>BUCK = 0x1 - Indicates the Buck is supplying the memory power domain. |

### 3.9.2.7 *BANDGAPEN Register*

**Band Gap Enable**

**OFFSET:** 0x000000FC

**INSTANCE 0 ADDRESS:** 0x400200FC

Band Gap Enable

**Table 118: BANDGAPEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | BGPEN |

**Table 119: BANDGAPEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:1 | RSVD | 0x0 | RO | RESERVED. |
| 0 | BGPEN | 0x0 | RW | Bandgap Enable<br><br>DIS = 0x0 - Bandgap disable.<br>EN = 0x1 - Bandgap enable. |

### 3.9.2.8 *SRAMPWDINSLEEP Register*

**Powerdown an SRAM Bank in Deep Sleep mode**

**OFFSET:** 0x00000140

**INSTANCE 0 ADDRESS:** 0x40020140

Powerdown an SRAM Bank in Deep Sleep mode

**Table 120: SRAMPWDINSLEEP Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | BANK7 | BANK6 | BANK5 | BANK4 | BANK3 | BANK2 | BANK1 | BANK0 |

### Table 121: SRAMPWDINSLEEP Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED. |
| 7 | BANK7 | 0x0 | RW | Force SRAM Bank 7 to powerdown in deep sleep mode, causing the contents of the bank to be lost.<br>NORMAL = 0x0 - SRAM Bank 7 normal operation.<br>PWRDN_IN_DEEPSLEEP = 0x1 - SRAM Bank 7 deep sleep. |
| 6 | BANK6 | 0x0 | RW | Force SRAM Bank 6 to powerdown in deep sleep mode, causing the contents of the bank to be lost.<br>NORMAL = 0x0 - SRAM Bank 6 normal operation.<br>PWRDN_IN_DEEPSLEEP = 0x1 - SRAM Bank 6 deep sleep. |
| 5 | BANK5 | 0x0 | RW | Force SRAM Bank 5 to powerdown in deep sleep mode, causing the contents of the bank to be lost.<br>NORMAL = 0x0 - SRAM Bank 5 normal operation.<br>PWRDN_IN_DEEPSLEEP = 0x1 - SRAM Bank 5 deep sleep. |
| 4 | BANK4 | 0x0 | RW | Force SRAM Bank 4 to powerdown in deep sleep mode, causing the contents of the bank to be lost.<br>NORMAL = 0x0 - SRAM Bank 4 normal operation.<br>PWRDN_IN_DEEPSLEEP = 0x1 - SRAM Bank 4 deep sleep. |
| 3 | BANK3 | 0x0 | RW | Force SRAM Bank 3 to powerdown in deep sleep mode, causing the contents of the bank to be lost.<br>NORMAL = 0x0 - SRAM Bank 3 normal operation.<br>PWRDN_IN_DEEPSLEEP = 0x1 - SRAM Bank 3 deep sleep. |
| 2 | BANK2 | 0x0 | RW | Force SRAM Bank 2 to powerdown in deep sleep mode, causing the contents of the bank to be lost.<br>NORMAL = 0x0 - SRAM Bank 2 normal operation.<br>PWRDN_IN_DEEPSLEEP = 0x1 - SRAM Bank 2 deep sleep. |
| 1 | BANK1 | 0x0 | RW | Force SRAM Bank 1 to powerdown in deep sleep mode, causing the contents of the bank to be lost.<br>NORMAL = 0x0 - SRAM Bank 1 normal operation.<br>PWRDN_IN_DEEPSLEEP = 0x1 - SRAM Bank 1 deep sleep. |
| 0 | BANK0 | 0x0 | RW | Force SRAM Bank 0 to powerdown in deep sleep mode, causing the contents of the bank to be lost.<br>NORMAL = 0x0 - SRAM Bank 0 normal operation.<br>PWRDN_IN_DEEPSLEEP = 0x1 - SRAM Bank 0 deep sleep. |

### 3.9.2.9 *SRAMPWRDIS Register*

**Disables individual banks of the SRAM array**

**OFFSET:** 0x00000144

**INSTANCE 0 ADDRESS:** 0x40020144

Disables individual banks of the SRAM array

**Table 122: SRAMPWRDIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | BANK7 | BANK6 | BANK5 | BANK4 | BANK3 | BANK2 | BANK1 | BANK0 |

**Table 123: SRAMPWRDIS Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED. |
| 7 | BANK7 | 0x0 | RW | Remove power from SRAM Bank 7 which will cause an access to its address space to generate a Hard Fault.<br>DIS = 0x1 - Disable SRAM Bank 7. |
| 6 | BANK6 | 0x0 | RW | Remove power from SRAM Bank 6 which will cause an access to its address space to generate a Hard Fault.<br>DIS = 0x1 - Disable SRAM Bank 6. |
| 5 | BANK5 | 0x0 | RW | Remove power from SRAM Bank 5 which will cause an access to its address space to generate a Hard Fault.<br>DIS = 0x1 - Disable SRAM Bank 5. |
| 4 | BANK4 | 0x0 | RW | Remove power from SRAM Bank 4 which will cause an access to its address space to generate a Hard Fault.<br>DIS = 0x1 - Disable SRAM Bank 4. |
| 3 | BANK3 | 0x0 | RW | Remove power from SRAM Bank 3 which will cause an access to its address space to generate a Hard Fault.<br>DIS = 0x1 - Disable SRAM Bank 3. |
| 2 | BANK2 | 0x0 | RW | Remove power from SRAM Bank 2 which will cause an access to its address space to generate a Hard Fault.<br>DIS = 0x1 - Disable SRAM Bank 2. |
| 1 | BANK1 | 0x0 | RW | Remove power from SRAM Bank 1 which will cause an access to its address space to generate a Hard Fault.<br>DIS = 0x1 - Disable SRAM Bank 1. |

**Table 123: SRAMPWRDIS Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 0 | BANK0 | 0x0 | RW | Remove power from SRAM Bank 0 which will cause an access to its address space to generate a Hard Fault.<br>DIS = 0x1 - Disable SRAM Bank 0. |

### 3.9.2.10 FLASHPWRDIS Register

**Disables individual banks of the Flash array**

**OFFSET:** 0x00000148

**INSTANCE 0 ADDRESS:** 0x40020148

Disables individual banks of the Flash array

**Table 124: FLASHPWRDIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | BANK1 | BANK0 |

**Table 125: FLASHPWRDIS Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:2 | RSVD | 0x0 | RO | RESERVED. |
| 1 | BANK1 | 0x0 | RW | Remove power from Flash Bank 1 which will cause an access to its address space to generate a Hard Fault.<br>DIS = 0x1 - Disable Flash instance 1. |
| 0 | BANK0 | 0x0 | RW | Remove power from Flash Bank 0 which will cause an access to its address space to generate a Hard Fault.<br>DIS = 0x1 - Disable Flash instance 0. |

### 3.9.2.11 ICODEFAULTADDR Register

**ICODE bus address which was present when a bus fault occurred.**

**OFFSET:** 0x000001C0

**INSTANCE 0 ADDRESS:** 0x400201C0

ICODE bus address which was present when a bus fault occurred.

**Table 126: ICODEFAULTADDR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 127: ICODEFAULTADDR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | ADDR | 0x0 | RO | The ICODE bus address observed when a Bus Fault occurred. Once an address is captured in this field, it is held until the corresponding Fault Observed bit is cleared in the FAULTSTATUS register. |

### 3.9.2.12 DCODEFAULTADDR Register

**DCODE bus address which was present when a bus fault occurred.**

**OFFSET:** 0x000001C4

**INSTANCE 0 ADDRESS:** 0x400201C4

DCODE bus address which was present when a bus fault occurred.

**Table 128: DCODEFAULTADDR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 129: DCODEFAULTADDR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | ADDR | 0x0 | RO | The DCODE bus address observed when a Bus Fault occurred. Once an address is captured in this field, it is held until the corresponding Fault Observed bit is cleared in the FAULTSTATUS register. |

### *3.9.2.13 SYSFAULTADDR Register*

**System bus address which was present when a bus fault occurred.**

**OFFSET:** 0x000001C8

**INSTANCE 0 ADDRESS:** 0x400201C8

System bus address which was present when a bus fault occurred.

**Table 130: SYSFAULTADDR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADDR | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 131: SYSFAULTADDR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | ADDR | 0x0 | RO | SYS bus address observed when a Bus Fault occurred. Once an address is captured in this field, it is held until the corresponding Fault Observed bit is cleared in the FAULTSTATUS register. |

### *3.9.2.14 FAULTSTATUS Register*

**Reflects the status of the bus decoders' fault detection. Any write to this register will clear all of the status bits within the register.**

**OFFSET:** 0x000001CC

**INSTANCE 0 ADDRESS:** 0x400201CC

Reflects the status of the bus decoders' fault detection. Any write to this register will clear all of the status bits within the register.

**Table 132: FAULTSTATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | SYS | DCODE | ICODE |

**Table 133: FAULTSTATUS Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:3 | RSVD | 0x0 | RO | RESERVED. |

**Table 133: FAULTSTATUS Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 2 | SYS | 0x0 | RW | SYS Bus Decoder Fault Detected bit. When set, a fault has been detected, and the SYSFAULTADDR register will contain the bus address which generated the fault.<br>NOFAULT = 0x0 - No bus fault has been detected.<br>FAULT = 0x1 - Bus fault detected. |
| 1 | DCODE | 0x0 | RW | DCODE Bus Decoder Fault Detected bit. When set, a fault has been detected, and the DCODEFAULTADDR register will contain the bus address which generated the fault.<br>NOFAULT = 0x0 - No DCODE fault has been detected.<br>FAULT = 0x1 - DCODE fault detected. |
| 0 | ICODE | 0x0 | RW | The ICODE Bus Decoder Fault Detected bit. When set, a fault has been detected, and the ICODEFAULTADDR register will contain the bus address which generated the fault.<br>NOFAULT = 0x0 - No ICODE fault has been detected.<br>FAULT = 0x1 - ICODE fault detected. |

### *3.9.2.15 FAULTCAPTUREEN Register*

**Enable the fault capture registers**

**OFFSET:** 0x000001D0

**INSTANCE 0 ADDRESS:** 0x400201D0

Enable the fault capture registers

**Table 134: FAULTCAPTUREEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | ENABLE |

**Table 135: FAULTCAPTUREEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:1 | RSVD | 0x0 | RO | RESERVED. |
| 0 | ENABLE | 0x0 | RW | Fault Capture Enable field. When set, the Fault Capture monitors are enabled and addresses which generate a hard fault are captured into the FAULTADDR registers.<br>DIS = 0x0 - Disable fault capture.<br>EN = 0x1 - Enable fault capture. |

### *3.9.2.16 TPIUCTRL Register*

**TPIU Control Register. Determines the clock enable and frequency for the M4's TPIU interface.**

**OFFSET:** 0x00000250

**INSTANCE 0 ADDRESS:** 0x40020250

TPIU Control Register. Determines the clock enable and frequency for the M4's TPIU interface.

**Table 136: TPIUCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | CLKSEL | | RSVD | | | | | | | ENABLE |

**Table 137: TPIUCTRL Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:10 | RSVD | 0x0 | RO | RESERVED. |
| 9:8 | CLKSEL | 0x0 | RW | This field selects the frequency of the ARM M4 TPIU port.<br><br>LOW_PWR = 0x0 - Low power state.<br>0MHz = 0x0 - Low power state.<br>6MHZ = 0x1 - Selects 6MHz frequency.<br>3MHZ = 0x2 - Selects 3MHz frequency.<br>1_5MHZ = 0x3 - Selects 1.5 MHz frequency. |
| 7:1 | RSVD | 0x0 | RO | RESERVED. |
| 0 | ENABLE | 0x0 | RW | TPIU Enable field. When set, the ARM M4 TPIU is enabled and data can be streamed out of the MCU's SWO port using the ARM ITM and TPIU modules.<br><br>DIS = 0x0 - Disable the TPIU.<br>EN = 0x1 - Enable the TPIU. |

## 3.10 Memory Subsystem

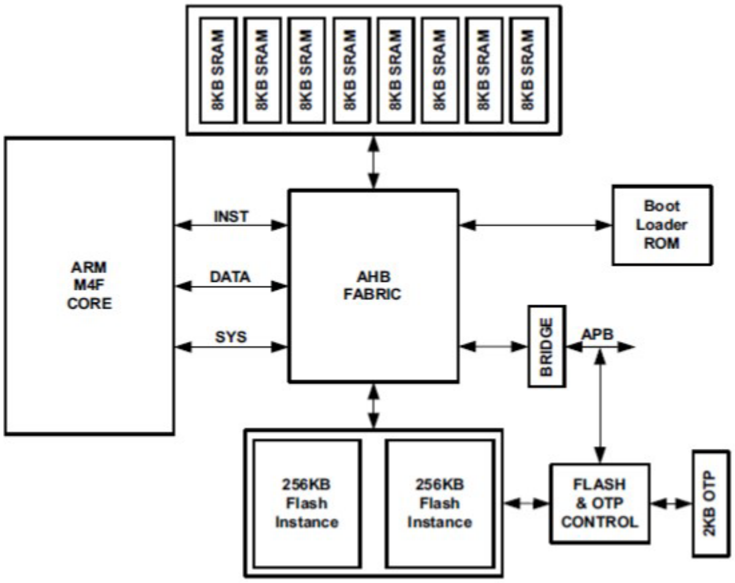


**Figure 4. Block Diagram for Flash and OTP Subsystem**

### *3.10.1 Features*

The Apollo SoC integrates four kinds of memory as shown in Figure 4:

- SRAM
- Flash
- Boot Loader ROM
- One Time Programmable (OTP) memory

Key features include:

- 64 KB SRAM
- Up to 2 Instances of 256 KB flash Memory
- 2048 Bytes OTP
  - 1024 Bytes contain factory preset per chip trim values.
  - 1024 Bytes for customer use, including flash protection fields
- Flash Protection specified in 16 KB Chunks
  - 32 OTP bits specify Write Protected Chunks
  - 32 OTP bits specify Read Protected Chunks
  - A Chunk is Execute Only if Both Corresponding Protection Bits Specified
  - OTP bits Specify Debugger Lock Out State
  - OTP bits Can Protect SRAM Contents From Debugger Inspection

### *3.10.2 Functional Overview*

The Apollo SoC Integrates up to 512 KB of on-board Flash non-volatile memory and 2048 bytes of one time programmable memory. These memories are managed by the APB flash controller for write operations.

During normal MCU code execution, the Flash Memory Controller translates requests from the CPU core to the Flash Memory instance for instruction and data fetches. The Controller is designed to return data in zero wait-states and can operate up to the maximum operating frequency of the CPU core.

The Flash Memory Controller facilitates flash erase and programming operations. When erase or programming operations are active, instructions cannot be fetched for execution from the Flash memory, so the on-chip SRAM would have to be used for code execution. To facilitate the management of Flash updates and OTP programming, a number of Flash helper functions are provided in the boot loader ROM.

The boot loader ROM contains2 instructions that are executed upon power up of the processor. Once a valid reset vector is establish at offset zero in the flash memory, the boot loader transfers control to users application by issuing a POR type reset which causes the core to enter the reset vector in flash. This process occupies less than 100 instructions in the boot loader. The remainder of the boot loader is occupied by a set of flash helper functions.

# 4. I$^2$C/SPI Master Module

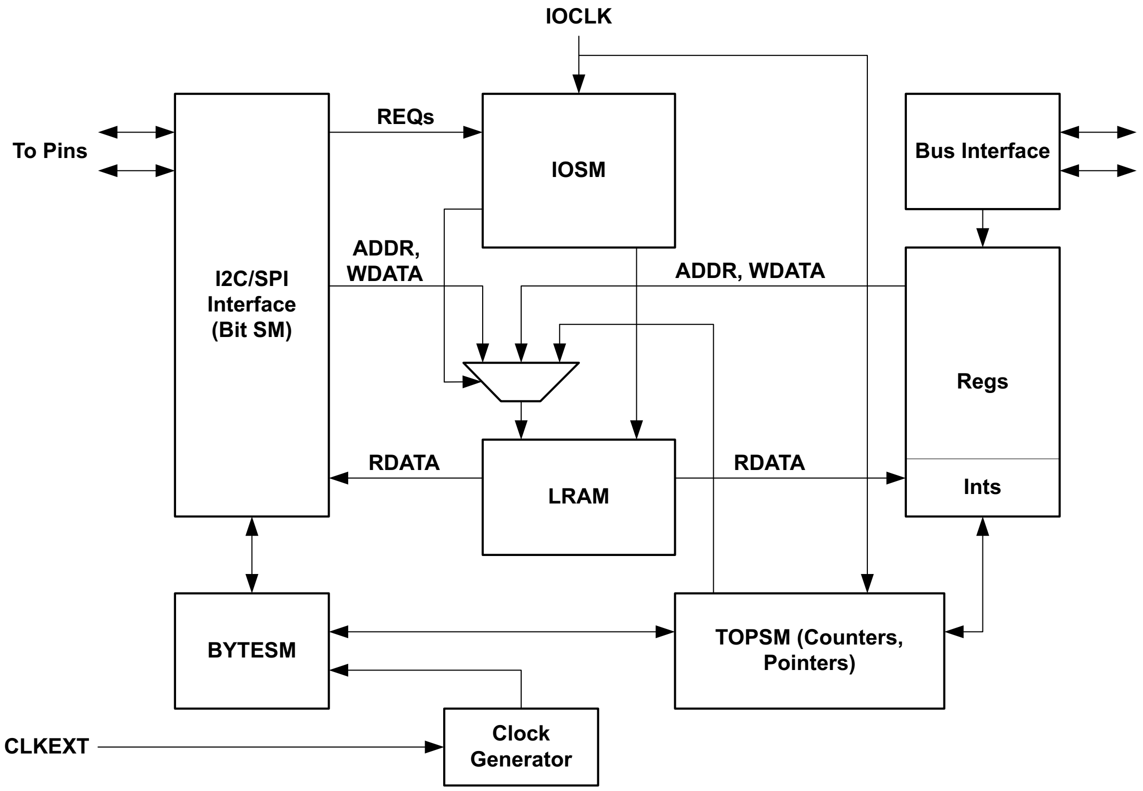


**Figure 5. Block Diagram for the I$^2$C/SPI Master Module**

## 4.1 Functional Overview

The Apollo SoC includes two I$^2$C/SPI Master Modules (IOM), shown in Figure 5, each of which functions as the master of an I$^2$C or SPI interface as selected by the IOMSTRn_IOMCFG_IFCSEL bit. A 64-byte bidirectional FIFO and a sophisticated Command mechanism allow simple initiation of I/O operations without requiring software interaction.

In I$^2$C mode the I$^2$C/SPI Master supports 7- and 10-bit addressing, multi-master arbitration, interface frequencies from 1.2 kHz to 1.0 MHz and up to 512-byte burst operations. In SPI mode the I$^2$C/SPI Master supports up to four slaves with automatic nCE selection, 3- and 4-wire implementation, all SPI polarity/phase combinations and up to 4095-byte burst operations, with both standard embedded address operations and raw read/write transfers. Interface timing limits are as specified in the Serial Peripheral Interface (SPI) Master Interface table of the Electrical Characteristics chapter.

## 4.2 Interface Clock Generation

The I$^2$C/SPI Master can generate a wide range of I/O interface clocks, as shown in Figure 6. The source clock is a scaled version of the HFRC 24 MHz clock, selected by IOMSTRn_CLKCFG_FSEL. A divide-by-3 circuit may be selected by IOMSTRn_CLKCFG_DIV3, which is particularly important in creating a useful SPI frequency of 8 MHz. The output of the divide-by-3 circuit may then be divided by an 8-bit value, IOMSTRn_CLKCFG_TOTPER + 1, to produce the interface clock. This structure allows very precise specification of the interface frequency, and produces a minimum available interface frequency of 1.2 kHz. If TOTPER division is enabled by IOMSTRn_CLKCFG_DIVEN, the length of the low period of the clock is specified by IOMSTRn_CLKCFG_LOWPER + 1. Otherwise, the clock will have a 50% duty cycle.

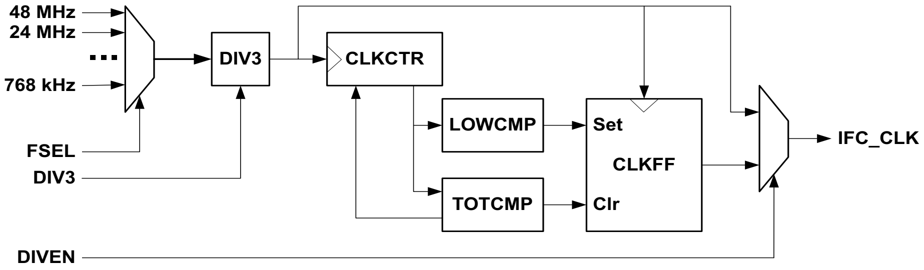


**Figure 6. I$^2$C/SPI Master Clock Generation**

## 4.3 Command Operation

In order to minimize the amount of time the CPU must be awake during I$^2$C/SPI Master operations, the architecture of the I$^2$C/SPI Master is organized around processing commands which transfer data to and from an internal 64-byte FIFO.

For writes to the interface, software fills the FIFO and then sends a single Command to the IOMSTRn_CMD Register. The Command includes either the I2C slave address or the SPI channel select, the desired address offset and the length of the transfer. At that point the I$^2$C/SPI Master executes the entire transfer, so the CPU can go to sleep. If more than 64 bytes are to be transferred, the Master will generate an interrupt when the FIFO crosses the write threshold IOMSTRn_FIFOTHR_FIFOWTHR so the CPU can wake up and refill the FIFO. The I$^2$C/SPI Master will generate an interrupt when the command is complete. In each case, the total number of bytes transferred in each operation is specified in the LENGTH field of the CMD Register.

For reads, the CMD Register is first written with the command and the CPU can go to sleep. The Master initiates the read and transfers read data to the FIFO. If the FIFO crosses the read threshold IOMSTRn_FIFOTHR_FIFORTHR, an interrupt is generated so the CPU can wake up and empty the FIFO. An interrupt is also generated when the command completes.

If the FIFO empties on a write or fills on a read, the I$^2$C/SPI Master will simply pause the interface clock until the CPU has read or written a byte from the FIFO. If software initiates an incorrect operation, such as attempting to read the FIFO on a write operation or when it is empty, or write the FIFO on a read operation or when it is full, the Master will generate an IACC error interrupt. If software attempts to write the Command Register when another Command is underway, the Master will generate an ICMD error interrupt.

## 4.4 FIFO

The I$^2$C/SPI Master includes a 64-byte local RAM (LRAM) for data transfers. The LRAM functions as a FIFO. Only 32-bit word accesses are supported to the FIFO from the CPU. When a write operation is underway, a word written to the FIFO will increment the IOMSTRn_FIFOPTR_FIFOSIZ register by 4 and decrement the IOMSTRn_FIFOPTR_FIFOREM register by 4. Reading a byte from the FIFO via the I/O interface decrements FIFOSIZ by 1 and increments FIFOREM by 1. When a read operation is underway, a word read from the FIFO decrements FIFOSIZ by 4 and increments FIFOREM by 4. A byte read from the I/O interface into the FIFO increments FIFOSIZ by 1 and decrements FIFOREM by 1. If FIFOSIZ becomes zero during a write operation or 0x40 on a read operation and there is more data to be transferred, the clock of the I/O interface is paused until software accesses the FIFO.

Two threshold registers, FIFORTHR and FIFOWTHR indicate when a THR interrupt should be generated to signal the processor that data should be transferred.

## 4.5 I$^2$C Interface

The I$^2$C/SPI Master supports a flexible set of Commands to implement a variety of standard I$^2$C operations. The I$^2$C interface consists of two lines: one bi-directional data line (SDA) and one clock line (SCL). Both the SDA and the SCL lines must be connected to a positive supply voltage via a pull-up resistor. By definition, a device that sends a message is called the “transmitter”, and the device that accepts the message is called the “receiver”. The device that controls the message transfer by driving SCL is called “master”. The devices that are controlled by the master are called “slaves”. The Apollo SoC I$^2$C Master is always a master device.

The following protocol has been defined:

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is high.
- Changes in the data line while the clock line is high will be interpreted as control signals.

A number of bus conditions have been defined (see Figure 7) and are described in the following sections

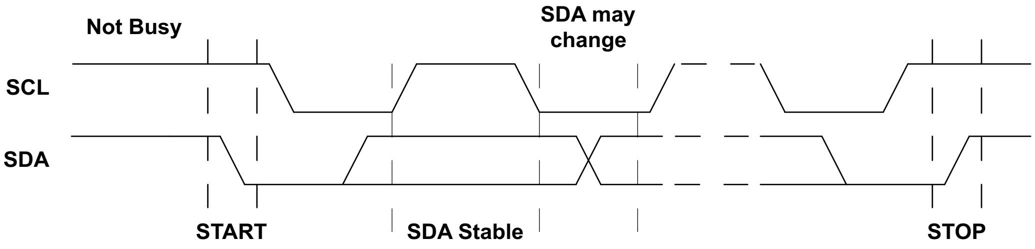


**Figure 7. Basic I$^2$C Conditions**

### 4.5.1 *Bus Not Busy*

Both SDA and SCL remain high.

### 4.5.2 *Start Data Transfer*

A change in the state of SDA from high to low, while SCL is high, defines the START condition. A START condition which occurs after a previous START, but before a STOP, is called a RESTART condition, and functions exactly like a normal STOP followed by a normal START.

### 4.5.3 Stop Data Transfer

A change in the state of SDA from low to high, while SCL is high, defines the STOP condition.

### 4.5.4 Data Valid

After a START condition, SDA is stable for the duration of the high period of SCL. The data on SDA may be changed during the low period of SCL. There is one clock pulse per bit of data. Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between the START and STOP conditions is not limited. The information is transmitted byte-wide and each receiver acknowledges with a ninth bit.

### 4.5.5 Acknowledge

Each byte of eight bits is followed by one acknowledge (ACK) bit as shown in Figure 8. This acknowledge bit is a low level driven onto SDA by the receiver, whereas the master generates an extra acknowledge related SCL pulse. A slave receiver which is addressed is obliged to generate an acknowledge after the reception of each byte. Also, on a read transfer, a master receiver must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is a stable low during the high period of the acknowledge related SCL pulse. A master receiver must signal an end-of-data to the slave transmitter by not generating an acknowledge (a NAK) on the last byte that has been clocked out of the slave. In this case, the transmitter must leave the data line high to enable the master to generate the STOP condition.

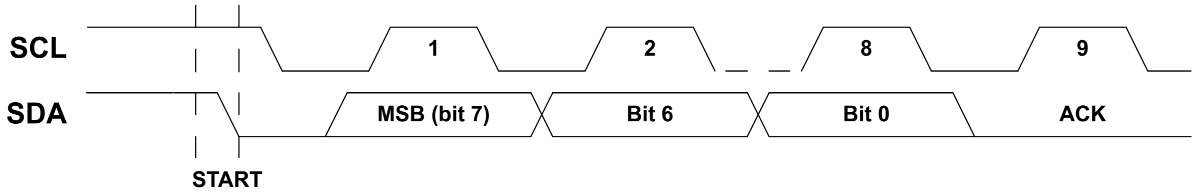


**Figure 8. I$^2$C Acknowledge**

### 4.5.6 I$^2$C Slave Addressing

For normal I$^2$C reads and writes, the Command specifies the address to be sent on the interface. Both 7-bit and 10-bit addressing are supported, as selected by 10BIT in the Command. The address is specified in the ADDRESS field.

Figure 9 shows the operation in 7-bit mode in which the master addresses the slave with a 7-bit address configured as 0xD0 in the lower 7 bits of the ADDRESS field. After the START condition, the 7-bit address is transmitted MSB first. If this address matches the lower 7 bits of an attached slave device, the eighth bit indicates a write (RW = 0) or a read (RW = 1) operation and the slave supplies the ACK. If no slave acknowledges the address, the transfer is terminated and a NAK error interrupt is generated.

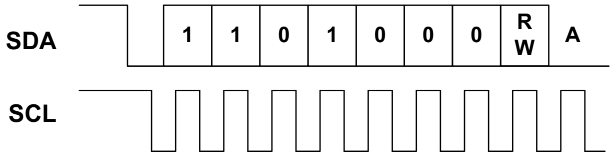


**Figure 9. I$^2$C 7-bit Address Operation**

Figure 10 shows the operation with which the master addresses the Apollo SoC with a 10-bit address configured at 0x536. After the START condition, the 10-bit preamble 0b11110 is transmitted first, followed by the upper two bits of the ADDRESS field and the eighth bit indicating a write (RW = 0) or a read (RW = 1) operation. If the upper two bits match the address of an attached slave device, it supplies the ACK. The next transfer includes the lower 8 bits of the ADDRESS field, and if these bits also match I2CADDR the slave again supplies the ACK. If no slave acknowledges either address byte, the transfer is terminated and a NAK error interrupt is generated.

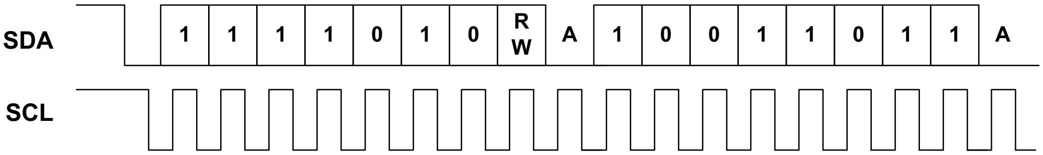


**Figure 10. I$^2$C 10-bit Address Operation**

### 4.5.7 *I$^2$C Offset Address Transmission*

If the OPER field of the CMD register selects a Normal Read or Write, the I$^2$C/SPI Master will first send an 8-bit Offset Address byte, where the offset is specified in the OFFSET field of CMD.

This transfer is shown in Figure 11. The Offset Address is loaded into the Address Pointer of the slave.

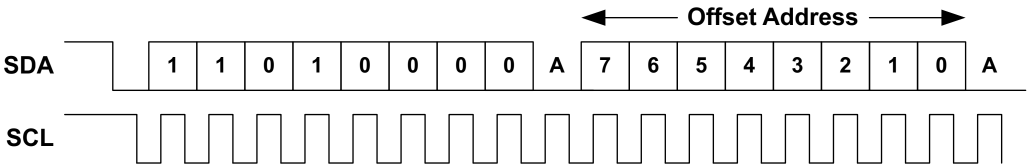


**Figure 11. I$^2$C Offset Address Transmission**

### 4.5.8 *I$^2$C Normal Write Operation*

In a Normal write operation the I$^2$C/SPI Master transmits to a slave receiver. The Address Operation has a RW value of 0, and the second byte contains the Offset Address, as in Figure 11. The next byte is written to the slave register selected by the Address Pointer (which was loaded with the Offset Address) and the Address Pointer is incremented. Subsequent transfers write bytes into successive registers until a STOP condition is received, as shown in Figure 12.

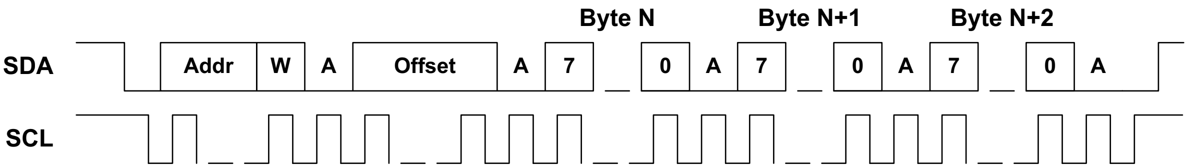


**Figure 12. I$^2$C Normal Write Operation**

### 4.5.9 I$^2$C Normal Read Operation

If a Normal Read operation is selected, the I$^2$C/SPI Master first executes an Offset Address Transmission to load the Address Pointer of the slave with the desired Offset Address.

A subsequent operation will again issue the address of the slave but with the RW bit as a 1 indicating a read operation. As shown in Figure 13, this transaction begins with a RESTART condition so that the interface will be held in a multi-master environment. After the address operation, the slave becomes the transmitter and sends the register value from the location pointed to by the Address Pointer, and the Address Pointer is incremented. Subsequent transactions produce successive register values, until the I$^2$C/SPI Master receiver responds with a NAK and a STOP to complete the operation.

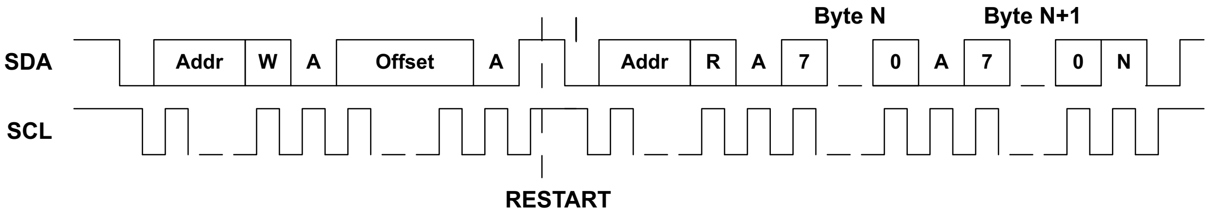


**Figure 13. I$^2$C Normal Read Operation**

### 4.5.10 I$^2$C Raw Write Operation

If a Raw Write is selected in the OPERfield of the CMD register, the I$^2$C/SPI Master does not execute the Offset Address Transmission, but simply begins transferring bytes as shown in Figure 14. This provides support for slave devices which do not implement the standard offset address architecture. The OFFSET field is not used in this case.

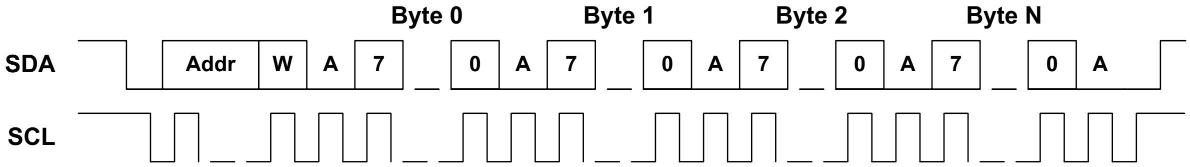


**Figure 14. I$^2$C Raw Write Operation**

### 4.5.11 I$^2$C Raw Read Operation

If a Raw Read is selected in the OPERfield of the CMD register, the I$^2$C/SPI Master does not execute the Offset Address Transmission, but simply begins transferring bytes with a read as shown in Figure 15. This is important for slave devices which do not support an Address Pointer architecture. For devices which do include an Address Pointer, multiple Raw Reads may be executed after a Normal Read to access subsequent registers as the Address Pointer increments, without having to execute the Offset Address Transmission for each access.

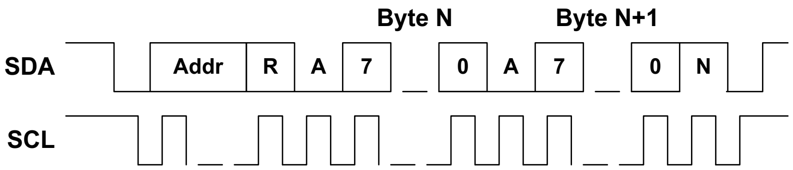


**Figure 15. I²C Raw Read Operation**

### *4.5.12 Holding the Interface with CONT*

In all of the previously described transactions, the I²C/SPI Master terminates the I²C operation with a STOP sequence. In environments where there are other masters connected to the I²C interface, it may be necessary for the Apollo SoC to hold the interface between Commands to insure that another master does not inadvertently access the same slave that the Apollo SoC is accessing. In order to implement this functionality, the CONT bit should be set in the CMD Register. This will cause the I²C/SPI Master to keep SDA high at the end of the transfer so that a STOP does not occur, and the next transaction begins with a RESTART instead of a START. Note that for a Normal Read the interface is held between the Offset Address Transmission and the actual read independent of the state of CONT, but it CONT is set the read transaction will not terminate with a STOP.

### *4.5.13 I²C Multi-master Arbitration*

The Apollo SoC I²C/SPI Master supports multi-master arbitration in I²C mode. There are two cases which must be handled.

The first is the case where another master initiates an I²C operation when the Apollo SoC Master is inactive. In this case the I²C/SPI Master will detect an I²C START operation on the interface and the START interrupt will be asserted, which tells the software not to generate any IO operations (which will not be executed in any case). Software then waits for the STOP interrupt, which reenables operation.

The second case is where another master initiates an operation at the same time as the Apollo SoC. In this case there will be a point where one master detects that it is not driving SDA low but the bus signal is low, and that master loses the arbitration to the other master. If the Apollo SoC I²C/SPI Master detects that it has lost arbitration, it will assert the ARB interrupt and immediately terminate its operation. Software must then wait for the STOP interrupt and re-execute the current Command.

## 4.6 SPI Operations

### *4.6.1 SPI Configuration*

The I²C/SPI Master supports all combinations of the polarity and phase modes of SPI using the IOMSTRn_IOMCFG_SPOL and IOMSTRn_IOMCFG_SPHA bits. It also may be configured in either 3-wire or 4-wire mode. In 4-wire mode, the MOSI and MISO interface signals use separate IO pins. In 3-wire mode, MOSI and MISO are multiplexed on a single IO pin for more efficient pin utilization. The 3/4 wire configuration is selected in the mapping function of the PINCFG module.

SPI operations may transfer up to 4095 bytes in a single transfer, as the UPLNGTH field of the CMD Register is appended to the standard LENGTH field to providea 12-bit length specification.

### 4.6.2 SPI Slave Addressing

In SPI mode, the Command specifies the slave channel to be used in the CHN field. The $I^2C$/SPI Master supports up to four slaves, each of which has its own nCE signal which can be configured on an IO pin. Additional slaves may be supported using GPIO pins and external decoding.

### 4.6.3 SPI Normal Write

Figure 16 shows the case of a SPI Normal Write with a one-byte address offset operation, whereby a write operation is selected in the OPER field. The operation is initiated when the $I^2C$/SPI Master pulls one of the nCE signals low. At that point the $I^2C$/SPI Master begins generating the clock on SCK and the offset address is transmitted from the master on the MOSI line, with the upper R/W bit of the offset field indicating read (if 0) or write (if 1). In this example the R/W bit is a one selecting a write operation. The entire offset byte, including the R/W bit, is taken from the OFFSET field of the CMD. This means that OFFSET[7] should be set to 0 if the slave expects a R/W bit. If the slave does not expect a R/W bit, this allows the first byte of a write to be completely specified in the OFFSET field, and a single byte write in that case can be executed without requiring any data to be loaded in to the FIFO.

Each subsequent byte is read from the FIFO and transmitted. The operation is terminated when the $I^2C$/SPI Master brings the nCE signal high. Note that the MISO line is not used in a write operation and is held in the high impedance state by the $I^2C$/SPI Master.

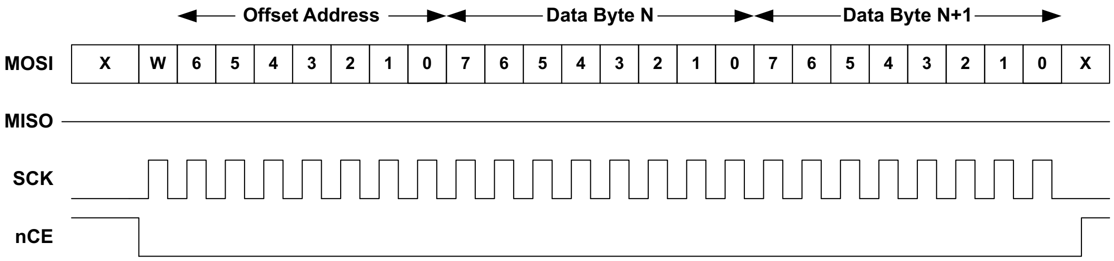


Figure 16. SPI Normal Write Operation (Single-byte Offset Address)

### 4.6.4 SPI Normal Read

Figure 17 shows the case of a Normal Read with a one-byte address offset operation, whereby a read operation is selected in the OPER field. The operation is initiated when the $I^2C$/SPI Master pulls one of the nCE signals low. At that point the $I^2C$/SPI Master begins driving the clock onto SCK and the address is transferred from the master to the slave just as it is in a write operation, but in this case the R/W bit is a 0 indicating a read. After the transfer of the last address bit (bit 0), the $I^2C$/SPI Master stops driving the MOSI line and begins loading the FIFO with the data on the MISO line. The transfer continues until the $I^2C$/SPI Master brings the nCE line high.

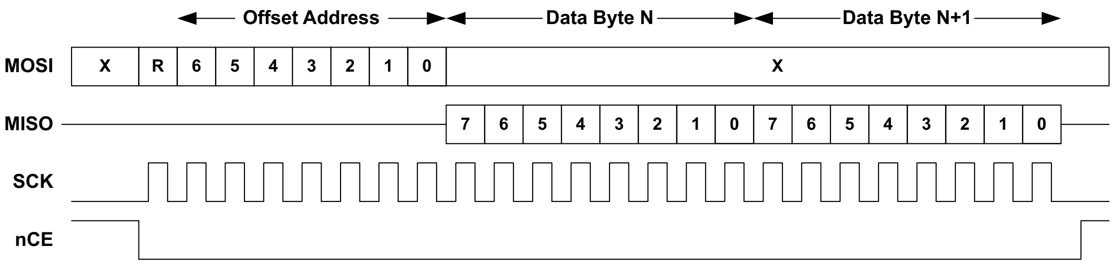


**Figure 17. SPI Normal Read Operation**

As with a Normal Write, the Offset Address byte including the R/W bit is taken from the offset field of CMD. If the slave expects a R/W bit, the MSB of the offset must be set accordingly. This allows reads from devices which have different formats for the address byte.

### *4.6.5 SPI Raw Write*

If a Raw Write is selected in the OPER field, the operation is similar to a Normal Write but the Offset Address byte is not sent and all data comes directly from the FIFO as shown in Figure 18. The OFFSET field is not used in this case.

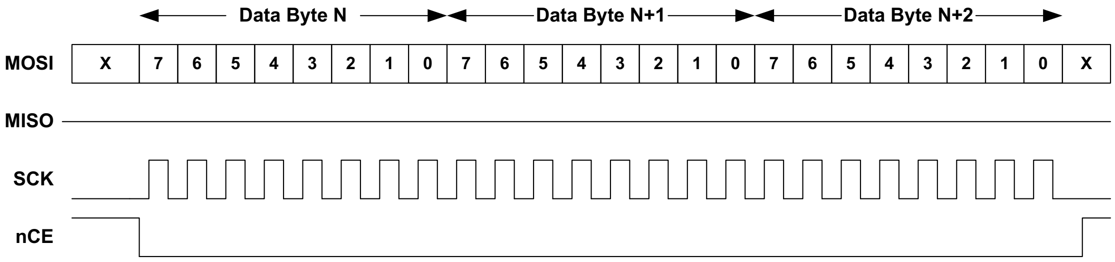


**Figure 18. SPI Raw Write Operation**

### *4.6.6 SPI Raw Read*

If a Raw Read is selected in the OPER field, the operation is simply the data transfer portion of a Normal Read. All data goes directly to the FIFO as shown in Figure 19. The OFFSET field is not used in this case.

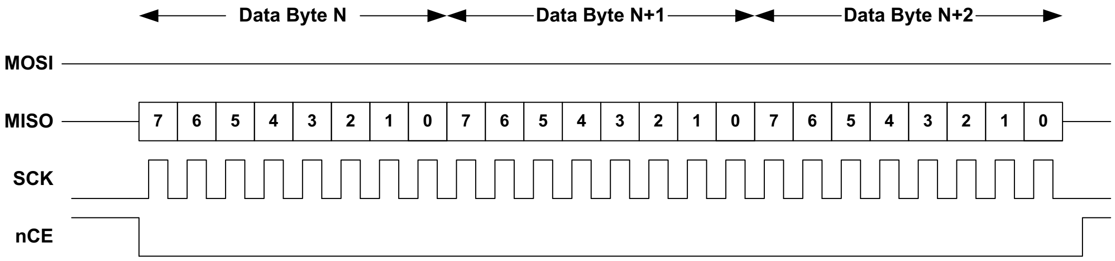


**Figure 19. SPI Raw Read Operation**

### 4.6.7 Complex SPI Operations

In some cases peripheral devices require more complex transaction sequences than those supported by a single Command. In order to support these transactions, the CONT bit may be set in the Command. In this case, the nCE pin selected by the Channel will remain asserted low at the end of the transaction, so that the next SPI operation will be seen as part of the same transaction. For example, there are peripheral devices which require both a Function and an Address Offset to be transmitted at the beginning of a read. Implementing this can be done in several ways. One example as shown in Figure 20 is:

1. Execute a Raw SPI write of length 2, with the data bytes being the Function and Offset. Set the CONT bit in this Command so nCE remains asserted low.
2. Execute a Raw SPI Read of the desired transfer length. The data will then be read into the FIFO. The CONT bit is not set in this Command.

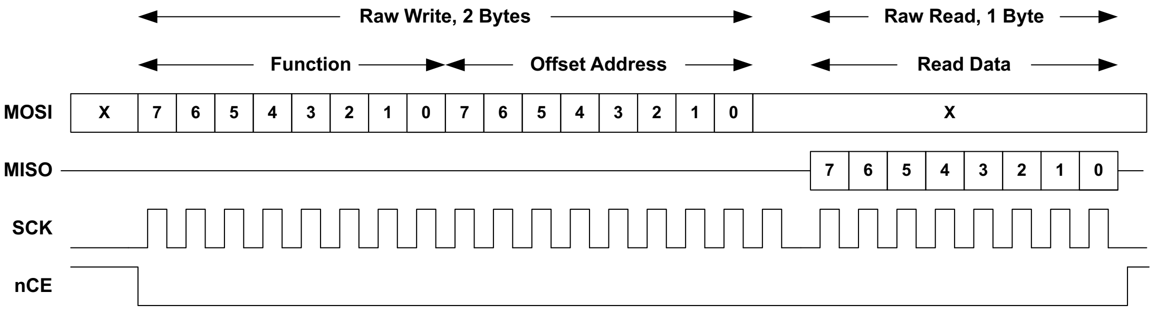


**Figure 20. SPI Combined Operation**

### 4.6.8 SPI Polarity and Phase

The Apollo SoC supports all combinations of CPOL (clock polarity) and CPHA (data phase) in SPI mode, as defined by the SPOL and SPHA bits. Figure 21 shows how these two bits affect the interface signal behavior.

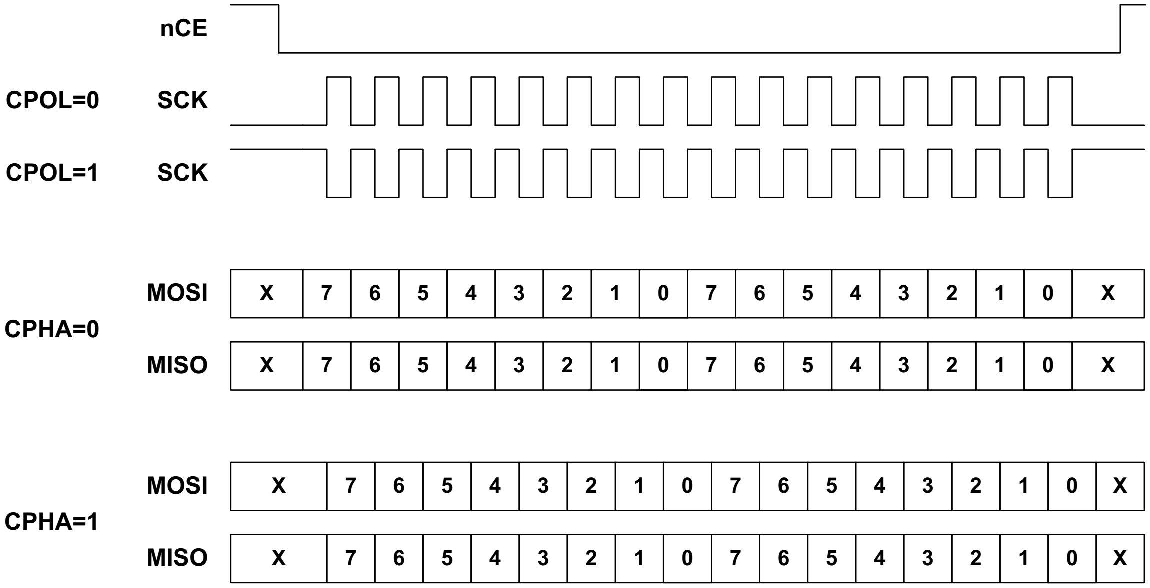

nCE
CPOL=0
SCK
CPOL=1
SCK
CPHA=0
MOSI
X 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 X
MISO
X 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 X
CPHA=1
MOSI
X 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 X
MISO
X 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 X

**Figure 21. SPI CPOL and CPHA**

If CPOL is 0, the clock SCK is normally low and positive pulses are generated during transfers. If CPOL is 1, SCK is normally high and negative pulses are generated during transfers.

If CPHA is 0, the data on the MOSI and MISO lines is sampled on the edge corresponding to the first SCK edge after nCE goes low (i.e. the rising edge if CPOL is 0 and the falling edge if CPOL is 1). Data on MISO and MOSI is driven on the opposite edge of SCK.

If CPHA is 1, the data on the MOSI and MISO lines is sampled on the edge corresponding to the second SCK edge after nCE goes low (i.e. the falling edge if CPOL is 0 and the rising edge if CPOL is 1). Data on MISO and MOSI is driven on the opposite edge of SCK.

The SPOL and SPHA bits may be changed between Commands if different slave devices have different requirements. In this case the IFCEN bit should be set to 0 either before or at the same time as SPHA and SPOL are changed, and then set back to 1 before CMD is written.

## 4.7 Bit Orientation

In both I$^2$C and SPI modes, the I$^2$C/SPI Master supports data transmission either LSB first or MSB first as configured by the LSB bit in the Command. If LSB is 0, data is transmitted and received MSB first. If LSB is 1, data is transmitted and received LSB first.

## 4.8 Minimizing Power

Each I$^2$C/SPI Master has an interface enable bit IOMSTRn_IOMCFG_IFCEN. This bit should be kept at 0 whenever the interface is not being used in order to minimize power consumption. The FIFO cannot be accessed if IFCEN is 0, although all of the other registers are accessible. When the module is not in use, the FSEL field should also be kept at 0 to minimize power. This is important even if IFCEN is disabled.

## 4.9 IOMSTR Registers

**I2C/SPI Master**

**INSTANCE 0 BASE ADDRESS:**0x50004000

**INSTANCE 1 BASE ADDRESS:**0x50005000

### *4.9.1 Register Memory Map*

**Table 138: IOMSTR Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0x50004000<br>0x50005000 | FIFO | FIFO Access Port |
| 0x50004100<br>0x50005100 | FIFOPTR | Current FIFO Pointers |
| 0x50004104<br>0x50005104 | TLNGTH | Transfer Length |
| 0x50004108<br>0x50005108 | FIFOTHR | FIFO Threshold Configuration |
| 0x5000410C<br>0x5000510C | CLKCFG | I/O Clock Configuration |
| 0x50004110<br>0x50005110 | CMD | Command Register |
| 0x50004114<br>0x50005114 | CMDRPT | Command Repeat Register |
| 0x50004118<br>0x50005118 | STATUS | Status Register |
| 0x5000411C<br>0x5000511C | CFG | I/O Master Configuration |
| 0x50004200<br>0x50005200 | INTEN | IO Master Interrupts: Enable |
| 0x50004204<br>0x50005204 | INTSTAT | IO Master Interrupts: Status |
| 0x50004208<br>0x50005208 | INTCLR | IO Master Interrupts: Clear |
| 0x5000420C<br>0x5000520C | INTSET | IO Master Interrupts: Set |

### 4.9.2 IOMSTR Registers

### 4.9.2.1 FIFO Register

**FIFO Access Port**

**OFFSET:** 0x00000000

**INSTANCE 0 ADDRESS:** 0x50004000

**INSTANCE 1 ADDRESS:** 0x50005000

FIFO Access Port

**Table 139: FIFO Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FIFO | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 140: FIFO Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | FIFO | 0x0 | RW | FIFO access port. |

### 4.9.2.2 FIFOPTR Register

**Current FIFO Pointers**

**OFFSET:** 0x00000100

**INSTANCE 0 ADDRESS:** 0x50004100

**INSTANCE 1 ADDRESS:** 0x50005100

Current FIFO Pointers

**Table 141: FIFOPTR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | FIFOREM | | | | | | | RSVD | | | | | | | | | FIFOSIZ | | | | | | |

**Table 142: FIFOPTR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:23 | RSVD | 0x0 | RO | RESERVED |

**Table 142: FIFOPTR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 22:16 | FIFOREM | 0x0 | RO | The number of bytes remaining in the FIFO (i.e. 64-FIFOSIZ). |
| 15:7 | RSVD | 0x0 | RO | RESERVED |
| 6:0 | FIFOSIZ | 0x0 | RO | The number of bytes currently in the FIFO. |

### 4.9.2.3 TLNGTH Register

**Transfer Length**

**OFFSET:** 0x00000104

**INSTANCE 0 ADDRESS:** 0x50004104

**INSTANCE 1 ADDRESS:** 0x50005104

Transfer Length

**Table 143: TLNGTH Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="20">RSVD</td><td colspan="12">TLNGTH</td></tr>
</table>

**Table 144: TLNGTH Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:12 | RSVD | 0x0 | RO | RESERVED |
| 11:0 | TLNGTH | 0x0 | RO | Remaining transfer length. |

### 4.9.2.4 FIFOTHR Register

**FIFO Threshold Configuration**

**OFFSET:** 0x00000108

**INSTANCE 0 ADDRESS:** 0x50004108

**INSTANCE 1 ADDRESS:** 0x50005108

FIFO Threshold Configuration

Table 145: FIFOTHR Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | FIFOWTHR | | | | | | RSVD | | FIFORTHR | | | | | |

Table 146: FIFOTHR Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:14 | RSVD | 0x0 | RO | RESERVED |
| 13:8 | FIFOWTHR | 0x0 | RW | FIFO write threshold. |
| 7:6 | RSVD | 0x0 | RO | RESERVED |
| 5:0 | FIFORTHR | 0x0 | RW | FIFO read threshold. |

### *4.9.2.5 CLKCFG Register*

**I/O Clock Configuration**

**OFFSET:** 0x0000010C

**INSTANCE 0 ADDRESS:** 0x5000410C

**INSTANCE 1 ADDRESS:** 0x5000510C

I/O Clock Configuration

Table 147: CLKCFG Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TOTPER | | | | | | | | LOWPER | | | | | | | | RSVD | | | DIVEN | DIV3 | FSEL | | | RSVD | | | | | | | |

Table 148: CLKCFG Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:24 | TOTPER | 0x0 | RW | Clock total count minus 1. |
| 23:16 | LOWPER | 0x0 | RW | Clock low count minus 1. |

**Table 148: CLKCFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12 | DIVEN | 0x0 | RW | Enable clock division by TOTPER.<br>DIS = 0x0 - Disable TOTPER division.<br>EN = 0x1 - Enable TOTPER division. |
| 11 | DIV3 | 0x0 | RW | Enable divide by 3.<br>DIS = 0x0 - Select divide by 1.<br>EN = 0x1 - Select divide by 3. |
| 10:8 | FSEL | 0x0 | RW | Select the input clock frequency.<br>HFRC_DIV64 = 0x0 - Selects the HFRC / 64 as the input clock.<br>HFRC = 0x1 - Selects the HFRC as the input clock.<br>HFRC_DIV2 = 0x2 - Selects the HFRC / 2 as the input clock.<br>HFRC_DIV4 = 0x3 - Selects the HFRC / 4 as the input clock.<br>HFRC_DIV8 = 0x4 - Selects the HFRC / 8 as the input clock.<br>HFRC_DIV16 = 0x5 - Selects the HFRC / 16 as the input clock.<br>HFRC_DIV32 = 0x6 - Selects the HFRC / 32 as the input clock.<br>RSVD = 0x7 - Reserved. |
| 7:0 | RSVD | 0x0 | RO | RESERVED |

### *4.9.2.6 CMD Register*

**Command Register**

**OFFSET:** 0x00000110

**INSTANCE 0 ADDRESS:** 0x50004110

**INSTANCE 1 ADDRESS:** 0x50005110

Command Register

**Table 149: CMD Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 150: CMD Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | CMD | 0x0 | RW | This register is the I/O Command.<br>POS_LENGTH = 0x0 - LSB bit position of the CMD LENGTH field.<br>POS_OFFSET = 0x8 - LSB bit position of the CMD OFFSET field.<br>POS_ADDRESS = 0x16 - LSB bit position of the I2C CMD ADDRESS field.<br>POS_CHNL = 0x16 - LSB bit position of the SPI CMD CHANNEL field.<br>POS_UPLNGTH = 0x23 - LSB bit position of the SPI CMD UPLNGTH field.<br>POS_10BIT = 0x26 - LSB bit position of the I2C CMD 10-bit field.<br>POS_LSB = 0x27 - LSB bit position of the CMD LSB-first field.<br>POS_CONT = 0x28 - LSB bit position of the CMD CONTinue field.<br>POS_OPER = 0x29 - LSB bit position of the CMD OPERation field.<br>MSK_LENGTH = 0xFF - LSB bit mask of the CMD LENGTH field.<br>MSK_OFFSET = 0xFF00 - LSB bit mask of the CMD OFFSET field.<br>MSK_ADDRESS = 0xFF0000 - LSB bit mask of the I2C CMD ADDRESS field.<br>MSK_CHNL = 0x70000 - LSB bit mask of the SPI CMD CHANNEL field.<br>MSK_UPLNGTH = 0x7800000 - LSB bit mask of the SPI CMD UPLNGTH field.<br>MSK_10BIT = 0x4000000 - LSB bit mask of the I2C CMD 10-bit field.<br>MSK_LSB = 0x8000000 - LSB bit mask of the CMD LSB-first field.<br>MSK_CONT = 0x10000000 - LSB bit mask of the CMD CONTinue field.<br>MSK_OPER = 0xE0000000 - LSB bit mask of the CMD OPERation field. |

### 4.9.2.7 *CMDRPT Register*

**Command Repeat Register**

**OFFSET:** 0x00000114

**INSTANCE 0 ADDRESS:** 0x50004114

**INSTANCE 1 ADDRESS:** 0x50005114

Command Repeat Register

**Table 151: CMDRPT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | CMDRPT | | | | |

**Table 152: CMDRPT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:5 | RSVD | 0x0 | RO | RESERVED |
| 4:0 | CMDRPT | 0x0 | RW | These bits hold the Command repeat count. |

#### 4.9.2.8 *STATUS Register*

**Status Register**

**OFFSET:** 0x00000118

**INSTANCE 0 ADDRESS:** 0x50004118

**INSTANCE 1 ADDRESS:** 0x50005118

Status Register

**Table 153: STATUS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | IDLEST | CMDACT | ERR |

**Table 154: STATUS Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:3 | RSVD | 0x0 | RO | RESERVED |
| 2 | IDLEST | 0x0 | RO | This bit indicates if the I/O state machine is IDLE.<br>IDLE = 0x1 - The I/O state machine is in the idle state. |
| 1 | CMDACT | 0x0 | RO | This bit indicates if the I/O Command is active.<br>ACTIVE = 0x1 - An I/O command is active. |
| 0 | ERR | 0x0 | RO | This bit indicates if an error interrupt has occurred.<br>ERROR = 0x1 - An error has been indicated by the IOM. |

#### *4.9.2.9 CFG Register*

**I/O Master Configuration**

**OFFSET:** 0x0000011C

**INSTANCE 0 ADDRESS:** 0x5000411C

**INSTANCE 1 ADDRESS:** 0x5000511C

I/O Master Configuration

**Table 155: CFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IFCEN | RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | SPHA | SPOL | IFCSEL |

**Table 156: CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | IFCEN | 0x0 | RW | This bit enables the IO Master.<br><br>DIS = 0x0 - Disable the IO Master.<br>EN = 0x1 - Enable the IO Master. |
| 30:3 | RSVD | 0x0 | RO | RESERVED |
| 2 | SPHA | 0x0 | RW | This bit selects SPI phase.<br><br>SAMPLE_LEADING_EDGE = 0x0 - Sample on the leading (first) clock edge.<br>SAMPLE_TRAILING_EDGE = 0x1 - Sample on the trailing (second) clock edge. |
| 1 | SPOL | 0x0 | RW | This bit selects SPI polarity.<br><br>CLK_BASE_0 = 0x0 - The base value of the clock is 0.<br>CLK_BASE_1 = 0x1 - The base value of the clock is 1. |
| 0 | IFCSEL | 0x0 | RW | This bit selects the I/O interface.<br><br>I2C = 0x0 - Selects I2C interface for the I/O Master.<br>SPI = 0x1 - Selects SPI interface for the I/O Master. |

### *4.9.2.10 INTEN Register*

**IO Master Interrupts: Enable**

**OFFSET:** 0x00000200

**INSTANCE 0 ADDRESS:** 0x50004200

**INSTANCE 1 ADDRESS:** 0x50005200

Set bits in this register to allow this module to generate the corresponding interrupt.

**Table 157: INTEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | ARB | STOP | START | ICMD | IACC | WTLEN | NAK | FOVFL | FUNDFL | THR | CMDCMP |

**Table 158: INTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:11 | RSVD | 0x0 | RO | RESERVED |
| 10 | ARB | 0x0 | RW | This is the arbitration loss interrupt. |
| 9 | STOP | 0x0 | RW | This is the STOP command interrupt. |
| 8 | START | 0x0 | RW | This is the START command interrupt. |
| 7 | ICMD | 0x0 | RW | This is the illegal command interrupt. |
| 6 | IACC | 0x0 | RW | This is the illegal FIFO access interrupt. |
| 5 | WTLEN | 0x0 | RW | This is the write length mismatch interrupt. |
| 4 | NAK | 0x0 | RW | This is the I2C NAK interrupt. |
| 3 | FOVFL | 0x0 | RW | This is the Read FIFO Overflow interrupt. |
| 2 | FUNDFL | 0x0 | RW | This is the Write FIFO Underflow interrupt. |
| 1 | THR | 0x0 | RW | This is the FIFO Threshold interrupt. |
| 0 | CMDCMP | 0x0 | RW | This is the Command Complete interrupt. |

### 4.9.2.11 *INTSTAT Register*

**IO Master Interrupts: Status**

**OFFSET:** 0x00000204

**INSTANCE 0 ADDRESS:** 0x50004204

**INSTANCE 1 ADDRESS:** 0x50005204

Read bits from this register to discover the cause of a recent interrupt.

**Table 159: INTSTAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | ARB | STOP | START | ICMD | IACC | WTLEN | NAK | FOVFL | FUNDFL | THR | CMDCMP |

**Table 160: INTSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:11 | RSVD | 0x0 | RO | RESERVED |
| 10 | ARB | 0x0 | RW | This is the arbitration loss interrupt. |
| 9 | STOP | 0x0 | RW | This is the STOP command interrupt. |
| 8 | START | 0x0 | RW | This is the START command interrupt. |
| 7 | ICMD | 0x0 | RW | This is the illegal command interrupt. |
| 6 | IACC | 0x0 | RW | This is the illegal FIFO access interrupt. |
| 5 | WTLEN | 0x0 | RW | This is the write length mismatch interrupt. |
| 4 | NAK | 0x0 | RW | This is the I2C NAK interrupt. |
| 3 | FOVFL | 0x0 | RW | This is the Read FIFO Overflow interrupt. |
| 2 | FUNDFL | 0x0 | RW | This is the Write FIFO Underflow interrupt. |
| 1 | THR | 0x0 | RW | This is the FIFO Threshold interrupt. |
| 0 | CMDCMP | 0x0 | RW | This is the Command Complete interrupt. |

### *4.9.2.12 INTCLR Register*

**IO Master Interrupts: Clear**

**OFFSET:** 0x00000208

**INSTANCE 0 ADDRESS:** 0x50004208

**INSTANCE 1 ADDRESS:** 0x50005208

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

**Table 161: INTCLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | ARB | STOP | START | ICMD | IACC | WTLEN | NAK | FOVFL | FUNDFL | THR | CMDCMP |

**Table 162: INTCLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:11 | RSVD | 0x0 | RO | RESERVED |
| 10 | ARB | 0x0 | RW | This is the arbitration loss interrupt. |
| 9 | STOP | 0x0 | RW | This is the STOP command interrupt. |
| 8 | START | 0x0 | RW | This is the START command interrupt. |
| 7 | ICMD | 0x0 | RW | This is the illegal command interrupt. |
| 6 | IACC | 0x0 | RW | This is the illegal FIFO access interrupt. |
| 5 | WTLEN | 0x0 | RW | This is the write length mismatch interrupt. |
| 4 | NAK | 0x0 | RW | This is the I2C NAK interrupt. |
| 3 | FOVFL | 0x0 | RW | This is the Read FIFO Overflow interrupt. |
| 2 | FUNDFL | 0x0 | RW | This is the Write FIFO Underflow interrupt. |
| 1 | THR | 0x0 | RW | This is the FIFO Threshold interrupt. |
| 0 | CMDCMP | 0x0 | RW | This is the Command Complete interrupt. |

### 4.9.2.13 INTSET Register

**IO Master Interrupts: Set**

**OFFSET:** 0x0000020C

**INSTANCE 0 ADDRESS:** 0x5000420C

**INSTANCE 1 ADDRESS:** 0x5000520C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

**Table 163: INTSET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | ARB | STOP | START | ICMD | IACC | WTLEN | NAK | FOVFL | FUNDFL | THR | CMDCMP |

**Table 164: INTSET Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:11 | RSVD | 0x0 | RO | RESERVED |
| 10 | ARB | 0x0 | RW | This is the arbitration loss interrupt. |
| 9 | STOP | 0x0 | RW | This is the STOP command interrupt. |
| 8 | START | 0x0 | RW | This is the START command interrupt. |
| 7 | ICMD | 0x0 | RW | This is the illegal command interrupt. |
| 6 | IACC | 0x0 | RW | This is the illegal FIFO access interrupt. |
| 5 | WTLEN | 0x0 | RW | This is the write length mismatch interrupt. |
| 4 | NAK | 0x0 | RW | This is the I2C NAK interrupt. |
| 3 | FOVFL | 0x0 | RW | This is the Read FIFO Overflow interrupt. |
| 2 | FUNDFL | 0x0 | RW | This is the Write FIFO Underflow interrupt. |
| 1 | THR | 0x0 | RW | This is the FIFO Threshold interrupt. |
| 0 | CMDCMP | 0x0 | RW | This is the Command Complete interrupt. |

# 5. I$^2$C/SPI Slave Module

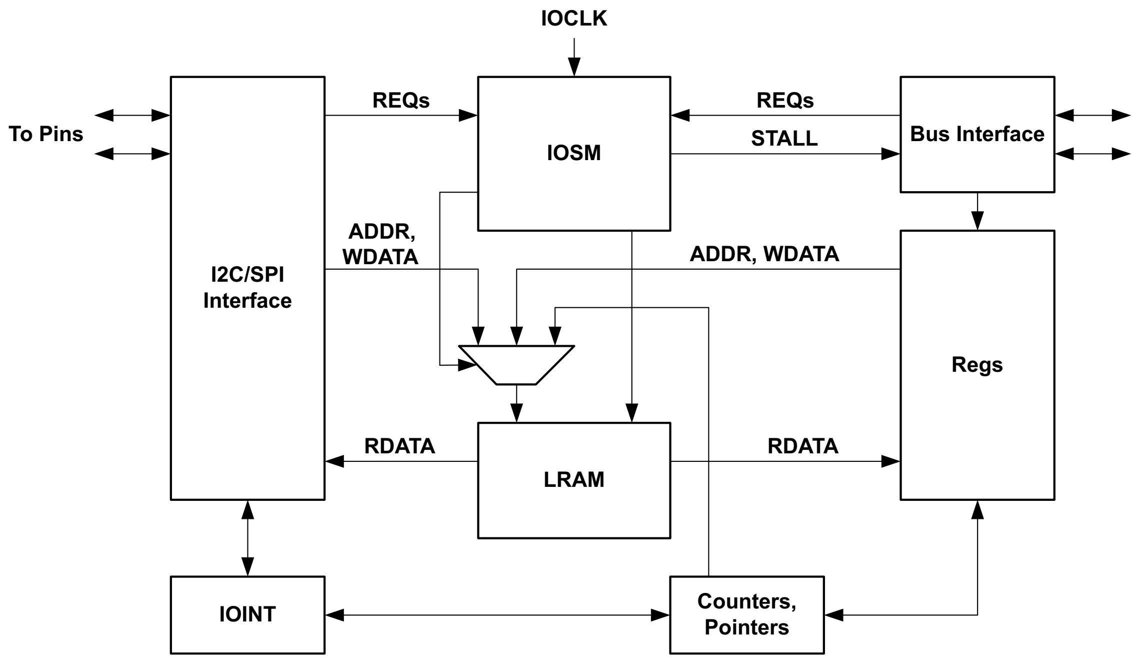


**Figure 22. Block diagram for the I$^2$C/SPI Slave Module**

## 5.1 Functional Overview

The I$^2$C/SPI Slave (IOS) Module, shown in Figure 22, allows the Apollo SoC to function as a Slave in an I$^2$C or SPI system. The I$^2$C/SPI Slave operates in an independent fashion, so that the device may be placed in a sleep mode and still receive operations over the I/O interface. The Slave may be configured to generate an interrupt on specific references.

The I$^2$C/SPI Slave contains 256 bytes of RAM which is only accessible when the module is enabled. This RAM may be flexibly configured into three spaces: a block directly accessible via the I/O interface, a block which functions as a FIFO for read operations on the interface, and a block of generally accessible RAM used to store parameters during deep sleep mode.

In I$^2$C mode the Slave supports fully configurable 7 and 10-bit addressing with interface timing limits as specified in the Inter-Integrated Circuit (I$^2$C) Interface section of the Electricals chapter. In SPI mode, the Slave supports all polarity/phase combinations and interface frequencies as specified in the Serial Peripheral Interface (SPI) Slave Interface section.

## 5.2 Local RAM Allocation

The $I^2C$/SPI Slave is built around a 256-byte local RAM (LRAM), through which all data flows between the CPU AHB and the IO interface. The $I^2C$/SPI Slave supports a 128-byte offset space when accessed from the I/O interface.

The LRAM is divided into three separate areas on 8-byte boundaries. These areas are:

1. A Direct Area for direct communication between the host and the MCU, which is mapped between the AHB address space and the I/O address space. This area is from LRAM address 0x00 to the address calculated from the 5-bit FIFOBASE field in the FIFO configuration register (FIFOCFG), minus 1. This 5-bit field (IOSLAVE_FIFOCFG_FIFOBASE) should contain a value that represents the start of the FIFO Area and, in so doing, defines the size of the Direct Area in 8-byte segments. Part of this area can be defined as IO Slave Read-only starting at any 8-byte segment defined by IOSLAVE_FIFOCFG_ROBASE and extending through the end of the Direct Area at FIFOBASE*8-1.
2. A FIFO Area which is used to stream data from the Apollo SoC. This memory is directly addressed from the AHB, but accessed from the I/O Interface using a single I/O address 0x7F as a streaming port. The FIFO area is from the LRAM address calculated from the value in the FIFOBASE field, FIFOBASE*8, to the LRAM address calculated from the value in the FIFOMAX field of the FIFOCFG register, IOSLAVE_FIFOCFG_FIFOMAX. The upper FIFO Area address is FIFOMAX*8-1. The maximum value for FIFOMAX is 0x20, which would result in an upper FIFO Area address of 0xFF.
3. A RAM Area which is accessible only from the AHB Slave. The RAM area is from the LRAM address calculated from the value in the FIFOMAX field of the FIFOCFG register, IOSLAVE_FIFOCFG_FIFOMAX, to address 0xFF. Setting FIFOMAX to 0x20 would result in a RAM area of zero size.

Figure 23 below shows the LRAM address mapping between the I/O interface and the AHB.

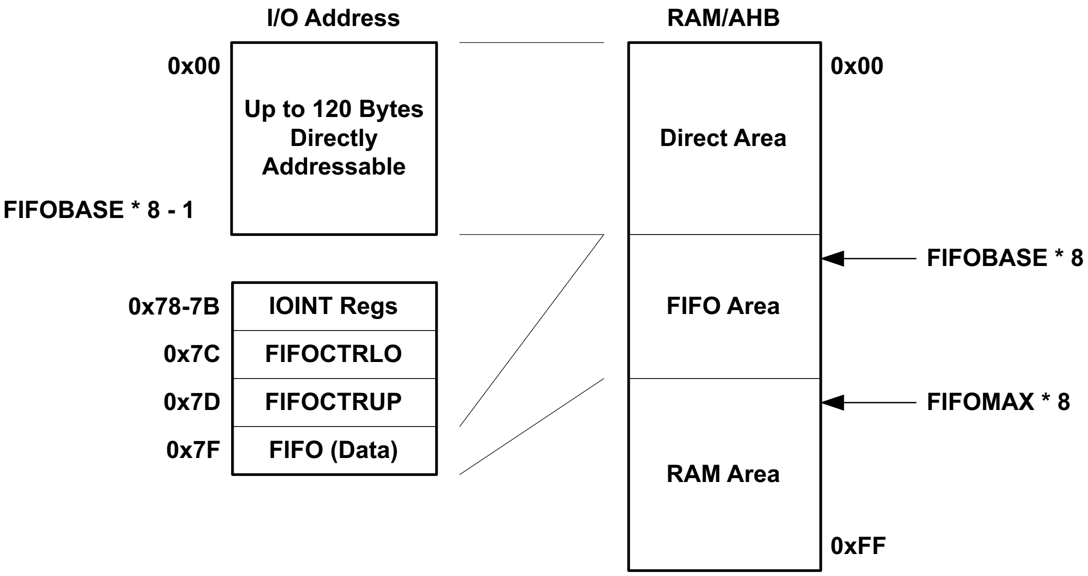


**Figure 23. $I^2C$/SPI Slave Module LRAM Addressing**

## 5.3 Direct Area Functions

The Direct Area is used for direct communications between the interface Host and the Apollo SoC. The Host may write a register in this Register Access space, called REGACC, and read it back without requiring the CPU to wake up, so that very low power interactions are supported. In some cases, however, accesses require interaction with the CPU.

REGACC interrupts are mapped in the Direct Area and operate as follows. Each REGACC interrupt status bit will be set whenever there is a read or write over the I$^2$C or SPI interface in the Direct Area with an offset address which corresponds to a particular REGACC interrupt. Table 165 below lists the offsets to memory locations within the Direct Area and corresponding interrupt bit settings in the REGACCINTSTAT register.

I/O writes to locations 0x0-0xF will set a corresponding interrupt flag in the REGACCINTSTAT register. These locations are typically used for specific commands to the Apollo SoC. Note that not all flags need generate an actual interrupt, so small multi-byte commands may be transmitted in this area. For example, a write to location 0x0 will set bit 31 of the REGACCINTSTAT register, a write to location 0x1 will set bit 30 of REGACCINTSTAT, and a write to location 0xF will set bit 16 of the REGACCINTSTAT register.

The upper 16 REGACC interrupts are each generated on an access to the last byte of a 32-bit word, starting at 0x10. I/O writes to locations 0x10 to 0x4F will set a corresponding interrupt flag in the REGACCINTSTAT register if the I/O address modulo 4 is 3 (i.e. addresses 0x13, 0x17, 0x1B, etc.). This allows larger transfers to be sent in a burst with a trigger being generated on the last write, and it also allows specifying a data buffer of any whole word size and have an interrupt generated on access to the last byte of the buffer. For example, a write to location 0x13 will set bit 15 of the REGACCINTSTAT register, a write to location 0x17 will set bit 14 of REGACCINTSTAT, and a write to location 0x4F will set bit 0 of the REGACCINTSTAT register.

Table 165 lists the offsets to memory locations within the Direct Address Space and corresponding interrupt bit settings in the REGACCINTSTAT register.

**Table 165: Mapping of Direct Area Access Interrupts and Corresponding REGACCINTSTAT Bits**

| REGACCINTSTAT Bit | Direct Area Offset Address |
|---|---|
| 31 | 0x0 |
| 30 | 0x1 |
| 29 | 0x2 |
| 28 | 0x3 |
| 27 | 0x4 |
| 26 | 0x5 |
| 25 | 0x6 |
| 24 | 0x7 |
| 23 | 0x8 |
| 22 | 0x9 |
| 21 | 0xA |
| 20 | 0xB |
| 19 | 0xC |
| 18 | 0xD |

**Table 165: Mapping of Direct Area Access Interrupts and Corresponding REGACCINTSTAT Bits**

| REGACCINTSTAT Bit | Direct Area Offset Address |
|---|---|
| 17 | 0xE |
| 16 | 0xF |
| 15 | 0x13 |
| 14 | 0x17 |
| 13 | 0x1B |
| 12 | 0x1F |
| 11 | 0x23 |
| 10 | 0x27 |
| 9 | 0x2B |
| 8 | 0x2F |
| 7 | 0x33 |
| 6 | 0x37 |
| 5 | 0x3B |
| 4 | 0x3F |
| 3 | 0x43 |
| 2 | 0x47 |
| 1 | 0x4B |
| 0 | 0x4F |

The REGACCINTSTAT register provides status of the 32 individual write interrupts. If an interrupt is enabled and set, it shows as a high bit in this register. The highest priority REGACC bit is bit 31 (set on access to address 0x00), and the lowest priority is bit 0 (set on access to address 0x4F).

The 5-bit IOSLAVE_PRENC register provides an encoded value of the highest priority of these interrupts to speed software decoding, and is therefore very useful for quickly servicing the highest priority REGACC interrupt (i.e. the one at the lowest offset address). The encoding works such that if interrupt 31 is set, PRENC will be 0. If interrupt 31 is not set and bit 30 is set, PRENC will be 1, and so on to the point where if bits 31-1 are not set and bit 0 is set PRENC will be 31. If no interrupts are set the value in PRENC is indeterminate.

The final special memory space within the Direct Area is a read-only area for the I/O Host, which is from I/O address (IOSLAVE_FIFOCFG_ROBASE * 8) to (FIFOBASE * 8 – 1). I/O writes to this address space will not change the LRAM, which allows the space to be used for returning status to the I/O Host. ROBASE should have a minimum value of 0x0A, representing a start address of 0x50 to allow space for special commands and burst writes in lower Direct Area space.

## 5.4 FIFO Area Functions

The FIFO is used to provide very efficient flow of data from the Apollo SoC to the I/O Host processor with minimal CPU interaction. A FIFO of up to 1023 bytes can be easily maintained by software, with the oldest bytes residing in the LRAM FIFO Area and the newer data being held in system SRAM and transferred to the $I^2C$/SPI Slave on demand. Several hardware features support this operation.

Figure 24 shows the basic FIFO operation. The main FIFO is held in a buffer in SRAM, and the oldest data in that FIFO has been transferred to the FIFO Area of the I/O Slave. The IOSLAVE_FIFOPTR_FIFOPTR register points to the next byte to be read on the I/O interface. IOSLAVE_FIFOPTR_FIFOSIZ holds the current number of valid bytes in the FIFO on the I$^2$C/SPI Slave, and FIFOCTR holds the total number of bytes in the FIFO. The value in IOSLAVE_FIFOCTR may be read indirectly at any time by the Host processor via the FIFOCTRUP_FIFOCTRLO registers to determine if there is FIFO data available (and how much is currently in the FIFO). I/O Host access to the FIFO counter is at offset 0x7C/D.

**WARNING**

The host read of the FIFOCTR value via FIFOCTRUP_FIFOCTRLO is not synchronized to the write clock. So if the host read happens during a FIFOCTR update (either through a read-modify-write of FIFOCTR register or an automatic update because of a write to the FIFOINC register by the Slave CPU), it is possible for the count value to be out of sync, impacting the value read in either or both the upper (FIFOCTRUP) and lower (FIFOCTRLO) bytes. This is a very rare case, but proper code would have the host read the two registers for the FIFOCTR value multiple times until consecutive reads are the same.

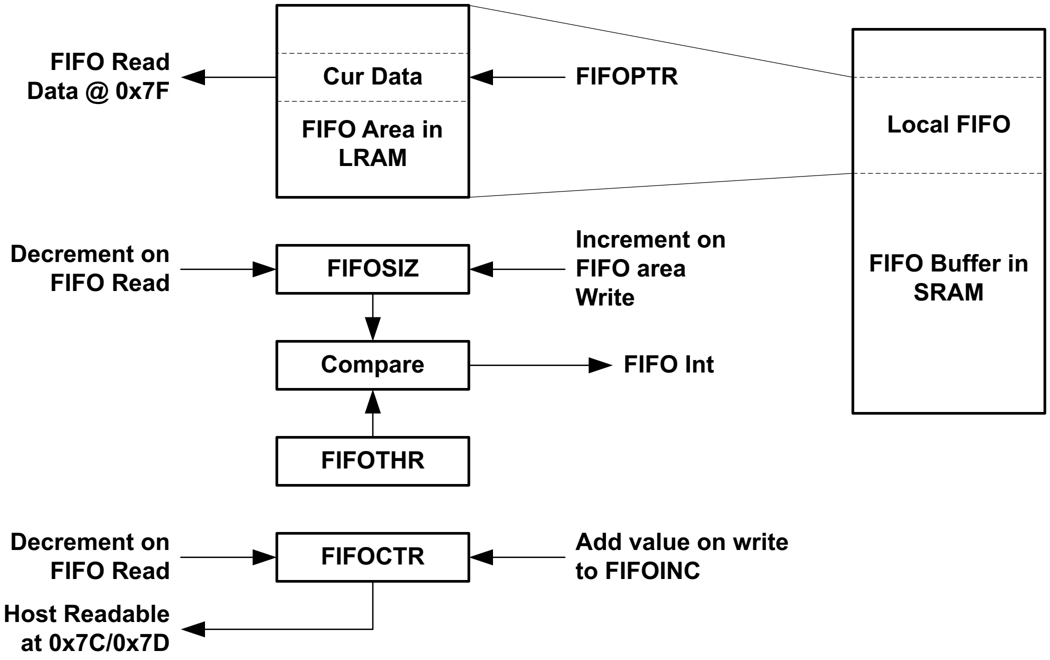


**Figure 24. I$^2$C/SPI Slave Module FIFO**

When the host reads a byte from the FIFO, the data retrieved is pointed to by FIFOPTR, FIFOPTR is incremented and wraps around in the FIFO Area if it reaches FIFOMAX. FIFOSIZ and FIFOCTR are each decremented by one. The Host can read FIFOCTR and then read that many bytes without further checking. Note that this process can continue without requiring a CPU wakeup. If the Host attempts to read the FIFO when FIFOSIZ is 0, the FUNDFL interrupt flag is set in both the I$^2$C Slave interrupt block and in the Host interrupt block.

When FIFOSIZ drops below the configured threshold IOSLAVE_FIFOTHR the FSIZE interrupt flag is set and if enabled an interrupt is sent to the CPU which will wake it up. At that point, the CPU can move as much data from the SRAM FIFO to the I$^2$C/SPI Slave FIFO as possible in a single operation and then go back to sleep. Since the FIFO Area can be quite large, CPU wake-ups will be very infrequent. If a write to the FIFOCTR which would increment the value beyond 1023 occurs, the FOVFL interrupt flag is set.

When some other process, such as a sensor read, produces new data for the FIFO, the CPU will add that data to the FIFO in SRAM, wrapping around as necessary. The IOSLAVE_FIFOINC register is then written with the number of bytes added to the FIFO, which is added to the FIFOCTR register in an atomic fashion. In this way the Host processor can always determine how much read data is available.

The FIFO interface offset 0x7F is treated uniquely by the I$^2$C/SPI Slave, in that an access to this address does not increment the Address Pointer. This allows the Host to initiate a burst read from address 0x7F of any length, and each read will supply the next byte in the FIFO.

## 5.5 Rearranging the FIFO

In normal operation the Host reads the oldest data from the FIFO, and the CPU writes new data onto the FIFO. In some cases it is desirable to modify this process, in particular for the FIFO to provide the newest data. The Apollo SoC supports such operation using a special control function.

If software desires to write the current sample to the front of the FIFO, it first checks the IOSLAVE_FUPD_IOREAD status bit to ensure that there is not a Host read operation from the FIFO underway. Once IOREAD is clear, software sets the IOSLAVE_FUPD_FIFOUPD bit, writes the new sample data to the front of the FIFO and modifies the FIFOPTR to point to the new data. At that point the FIFOUPD bit is cleared.

If the Host attempts a FIFO read operation while the FIFOUPD is set, a RDERR interrupt will be generated to the Host and the FRDERR interrupt flag will be set. The Host must either poll the RDERR interrupt bit at the end of each operation or configure a hardware interrupt. Note that if the software does not support alternate FIFO ordering, the Host does not have to check the RDERR function.

## 5.6 Interface Interrupts

The CPU may also signal the Host via the IOINT interrupt, which may be connected to an Apollo SoC pin and driven to the Host. Eight interrupts are available to be combined into the IOINT interrupt, and the Host can enable, read, clear and set these interrupts via the I/O interface. Software on the CPU can set 6 of the interrupts (SWINT0 through SWINT5) to communicate a variety of situations to the Host, and the other two interrupts indicate errors such as an attempt by the Host to read the FIFO when it is empty. A CPU interrupt is generated whenever the Host writes any IOINT registers (for example, to clear an interrupt) so the CPU can manage the interrupt interaction.

The I$^2$C/SPI Slave includes a mechanism to allow the Host CPU and the Apollo SoC to each interrupt the other via a set of eight interrupts. The Host CPU accesses these interrupts via interface locations 0x78-0x7B, and the Apollo accesses these interrupts in the IOINTCTL Register.

The Host CPU may enable or disable any of the eight interrupts by writing the corresponding bit in the IOINTEN field of the IOINTCTL Register, which is accessed by the Host at interface location 0x78. The Host CPU may then clear or set any of the interrupts by writing a 1 to the corresponding bit of the clear (at location 0x7A) or set (at location 0x7B) registers. The current state of all eight interrupts may be read in the IOINT field at location 0x79. Note that this structure is identical to the standard Apollo SoC interrupts in all modules. The MCU can read the value of the eight interrupt enables in the IOINTEN field of IOINTCTL, and can read the values of the eight interrupt status bits in the IOINT field of the IOINTCTL register. These two fields are read only. Table 166 summarizes these I/O interface interrupts and how they can be controlled and read.

**Table 166: I/O Interface Interrupt Control**

| RAM Location | IOINT Register[1] | Function | MCU Register Field | Description |
|---|---|---|---|---|
| 0x78 | IOINTEN | I/O Interrupt Enable | IOINTCTL_IOINTEN (R/O) | Each interrupt can be individually enabled by I/O Host, but can only be read by the MCU |
| 0x79 | IOINT | I/O Interrupt State | IOINTCTL_IOINT (R/O) | State of each interrupt, set or cleared, can be read by either the I/O Host or by the MCU |
| 0x7A | IOINTCLR | I/O Interrupt Clear | IOINTCTL_IOINTCLR (W/O | Each interrupt can be individually cleared by the I/O Host, but the MCU can (only) clear all of them at once |
| 0x7B | IOINTSET | I/O Interrupt Set | IOINTCTL_IOINTSET (W/O) | Each interrupt can be individually set by either the I/O Host or the MCU |

[1] Readable by the I/O Host

The Apollo SoC software may set any of the eight interrupt status register bits by writing a 1 to the corresponding bit of the IOINTSET field of the IOINTCTL Register, and may clear all of the interrupts by writing a 1 to the IOINTCLR bit of the IOINTCTL register. This allows the MCU to generate a software interrupt to the Host device. In addition, a FIFO underflow interrupt FUNDFL in the I$^2$C/SPI Slave will set interrupt bit 7, and a FIFO read error interrupt FRDERR will set interrupt bit 6 of the IO interrupt status register IOINT. Note that the MCU software cannot write the IOINTEN register, so that IO interrupts are controlled completely by the Host processor.

If any of the IOINT interrupt bits are set and the corresponding bit in IOINTEN is set, an IOINT interrupt will be generated. If the GPIO configuration registers have configured PAD4 as IOINT, that interrupt will be driven directly onto PAD_IO[4]. This pin should be connected to an interrupt input pin of the Host interface device so that it can receive the interrupt and service it.

If the Host device writes to any of the interrupt register access locations (any location in 0x78-0x7B) the IOINTW interrupt will be set in the I2C/SPI INTSTAT Register. This allows Apollo SoC software to receive a software interrupt from the Host device. Note that this interrupt will occur for all writes by the Host, including a write to clear an interrupt.

## 5.7 Host Address Space and Registers

The Host of the I/O interface can access 128 bytes in the I$^2$C/SPI Slave in either I$^2$C or SPI mode. Offsets 0x00 to 0x77 may be directly mapped to the Direct RAM Area. The remaining eight offset locations access hardware functions within the I$^2$C/SPI Slave. The R/W indicator is referring to accesses from the Host.

## 5.8 I$^2$C Interface

The Apollo SoC I$^2$C Slave interface operates as a standard slave. The device is accessed at an address configured in the IOSLAVE_IOSCFG_I2CADDR field, and supports Fast Mode Plus (up to 1 MHz). Both 7-bit and 10-bit address modes are supported, as selected by IOSLAVE_IOSCFG_10BIT. The I$^2$C interface consists of two lines: one bi-directional data line (SDA) and one clock line (SCL). Both the SDA and the SCL lines must be connected to a positive supply voltage via a pull-up resistor. By definition, a device that sends a message is called the “transmitter”, and the device that accepts the message is called the “receiver”. The device that controls the message transfer by driving SCL is called “master”. The devices that are controlled by the master are called “slaves”. The MCU’s I$^2$C Slave is always a slave device.

The following protocol has been defined:

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is high.
- Changes in the data line while the clock line is high will be interpreted as control signals.

A number of bus conditions have been defined (see Figure 25) and are described in the following sections.

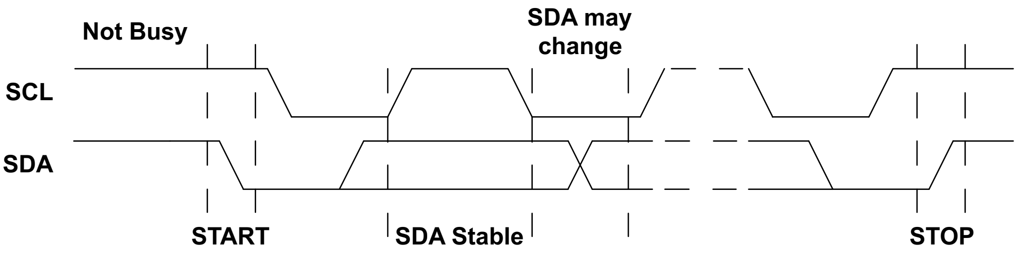


**Figure 25. Basic $I^2C$ Conditions**

### 5.8.1 Bus Not Busy

Both SDA and SCL remain high.

### 5.8.2 Start Data Transfer

A change in the state of SDA from high to low, while SCL is high, defines the START condition. A START condition which occurs after a previous START but before a STOP is called a RESTART condition, and functions exactly like a normal STOP followed by a normal START.

### 5.8.3 Stop Data Transfer

A change in the state of SDA from low to high, while SCL is high, defines the STOP condition.

### 5.8.4 Data Valid

After a START condition, SDA is stable for the duration of the high period of SCL. The data on SDA may be changed during the low period of SCL. There is one clock pulse per bit of data. Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between the START and STOP conditions is not limited. The information is transmitted byte-wide and each receiver acknowledges with a ninth bit.

### 5.8.5 Acknowledge

Each byte of eight bits is followed by one Acknowledge (ACK) bit as shown in Figure 26. This Acknowledge bit is a low level driven onto SDA by the receiver, whereas the master generates an extra ACK related SCL pulse. A slave receiver which is addressed is obliged to generate an Acknowledge after the reception of each byte. Also, on a read transfer a master receiver must generate an Acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges must pull down the SDA line during the Acknowledge clock pulse in such a way that the SDA line is a stable low during the high period of the Acknowledge related SCL pulse. A master receiver must signal an end-of-data to the slave transmitter by not generating an Acknowledge (a NAK) on the last byte that has been clocked out of the slave. In this case, the transmitter must leave the data line high to enable the master to generate the STOP condition.

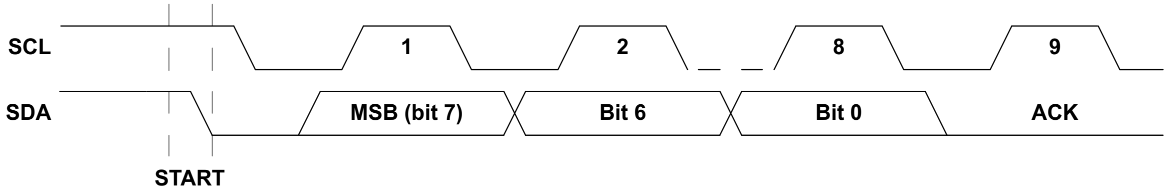


Figure 26. I$^2$C Acknowledge

### 5.8.6 *Address Operation*

In I$^2$C mode, the I$^2$C/SPI Slave supports either 7-bit or 10-bit addressing, selected by the 10BIT bit in the IOSCFG Register. Figure 27 shows the operation in 7-bit mode in which the master addresses the Apollo SoC with a 7-bit address configured as 0xD2 in the CFG_I2CADDR field. After the START condition, the 7-bit address is transmitted MSB first. If this address matches the lower 7 bits of the CFG_I2CADDR field, the MCU is selected, the eighth bit indicate a write (RW = 0) or a read (RW = 1) operation and the MCU supplies the ACK. The MCU ignores all other address values and does not respond with an ACK.

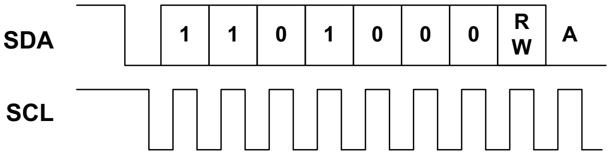


Figure 27. I$^2$C 7-bit Address Operation

Figure 28 shows the operation with which the master addresses the MCU with a 10-bit address configured at 0x536. After the START condition, the 10-bit preamble 0b11110 is transmitted first, followed by the first two address bits and the eighth bit indicating a write (RW = 0) or a read (RW = 1) operation. If the upper two bits match the I2CADDR value, the I$^2$C/SPI Slave supplies the ACK. The next transfer includes the lower 8 bits of the address, and if these bits also match I2CADDR the MCU again supplies the ACK. The I$^2$C/SPI Slave ignores all other address values and does not respond with an ACK.

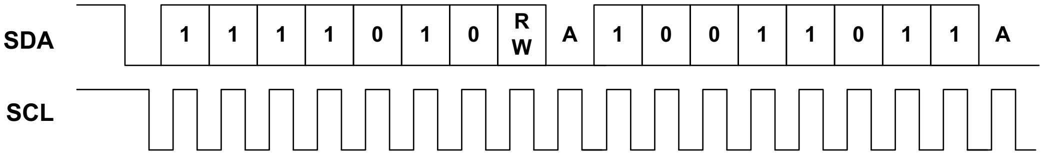


Figure 28. I$^2$C 10-bit Address Operation

### 5.8.7 *Offset Address Transmission*

If the RW bit of the Address Operation indicates a write, the next byte transmitted from the master is the Offset Address as shown in Figure 29. This value is loaded into the Address Pointer of the I$^2$C/SPI Slave.

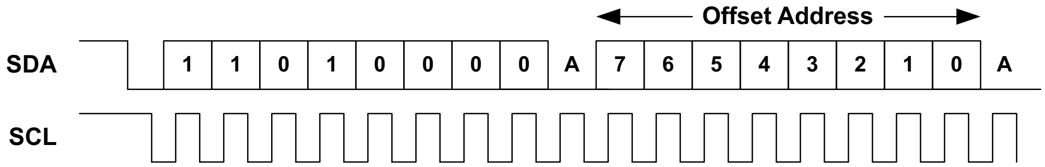


Figure 29. I$^2$C Offset Address Transmission

### 5.8.8 Write Operation

In a write operation the master transmitter transmits to the Apollo SoC slave receiver. The Address Operation has a RW value of 0, and the second byte contains the Offset Address as in Figure 29. The next byte is written to the register selected by the Address Pointer (which was loaded with the Offset Address) and the Address Pointer is incremented. Subsequent transfers write bytes into successive registers until a STOP condition is received, as shown in Figure 30. Note that if the Address Pointer is at 0x7F, it will not increment on the write.

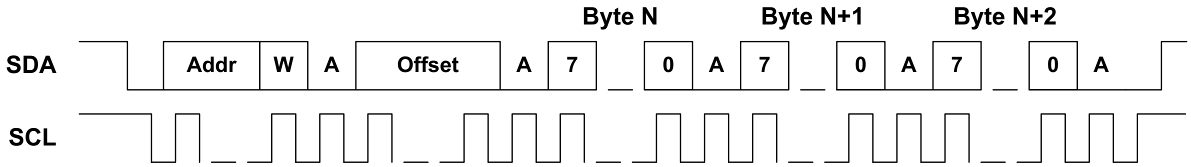


Figure 30. I$^2$C Write Operation

### 5.8.9 Read Operation

In a read operation, the master first executes an Offset Address Transmission to load the Address Pointer with the desired Offset Address. A subsequent operation will again issue the address of the MCU but with the RW bit as a 1 indicating a read operation. Figure 31 shows this transaction beginning with a RESTART condition, although a STOP followed by a START may also be used. After the address operation, the slave becomes the transmitter and sends the register value from the location pointed to by the Address Pointer, and the Address Pointer is incremented. Subsequent transactions produce successive register values, until the master receiver responds with a NAK and a STOP to complete the operation. Because the Address Pointer holds a valid register address, the master may initiate another read sequence at this point without performing another Offset Address operation. Note that if the Address Pointer is at 0x7F, it will not increment on the read.

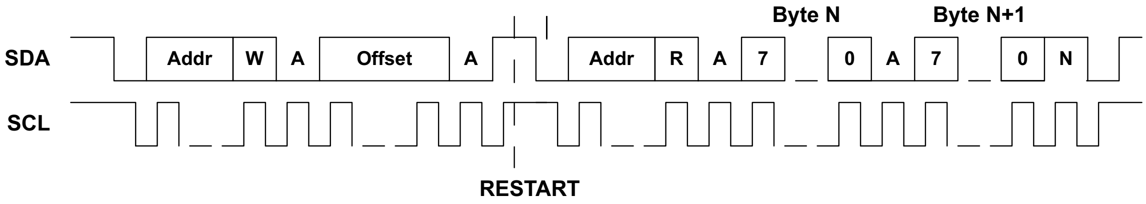


Figure 31. I$^2$C Read Operation

### 5.8.10 General Address Detection

The I$^2$C/SPI Slave may be configured to detect an I$^2$C General Address (0x00) write. If this address is detected, the first data byte written is stored in the IOSLAVE_GADATA Register and the GENAD interrupt flag is set. This allows software to create the appropriate response, which is typically to reset the I$^2$C/SPI Slave.

## 5.9 SPI Interface

The I$^2$C/SPI Slave includes a standard 3-wire or 4-wire SPI interface. The serial peripheral interface (SPI) bus is intended for synchronous communication between different ICs. 4-wire SPI consists of four signal lines: serial data input (MOSI), serial data output (MISO), serial clock (SCL) and an active low chip enable (nCE). The I$^2$C/SPI Slave may be connected to a master with a 3-wire SPI interface by configuring 3-wire mode in the pin configuration block of the GPIO module, which will tie MOSI and MISO together. By definition, a device that sends a message is called the “transmitter”, and the device that accepts the message is called the “receiver”. The device that controls the message transfer by driving SCL is called “master”. The devices that are controlled by the master are called “slaves”. The I$^2$C/SPI Slave SPI Slave is always a slave device.

The nCE input is used to initiate and terminate a data transfer. The SCL input is used to synchronize data transfer between the master and the slave devices via the MOSI (master to slave) and MISO (slave to master) lines. The SCL input, which is generated by the master, is active only during address and data transfer to any device on the SPI bus.

The I$^2$C/SPI Slave supports all SPI configurations of CPOL and CPHA using the SPOL configuration bit. There is one clock for each bit transferred. Address and data bits are transferred in groups of eight bits.

### 5.9.1 Write Operation

Figure 32 shows a SPI write operation. The operation is initiated when the nCE signal to the MCU goes low. At that point an 8-bit Address byte is transmitted from the master on the MOSI line, with the upper RW bit indicating read (if 0) or write (if 1). In this example the RW bit is a one selecting a write operation, and the lower 7 bits of the Address byte contain the Offset Address, which is loaded into the Address Pointer of the I$^2$C/SPI Slave.

Each subsequent byte is loaded into the register selected by the Address Pointer, and the Address Pointer is incremented. The operation is terminated by the master by bringing the nCE signal high. Note that the MISO line is not used in a write operation and is held in the high impedance state by the I$^2$C/SPI Slave. Note also that if the Address Pointer is 0x7F, it does not increment on the read.

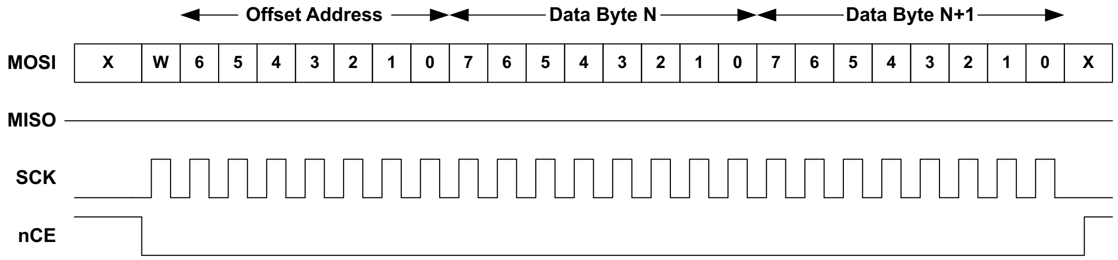


**Figure 32. SPI Write Operation**

### *5.9.2 Read Operation*

Figure 33 shows a read operation. The address is transferred from the master to the slave just as it is in a write operation, but in this case the RW bit is a 0 indicating a read. After the transfer of the last address bit (bit 0), the I$^2$C/SPI Slave begins driving data from the register selected by the Address Pointer onto the MISO line, bit 7 first, and the Address Pointer is incremented. The transfer continues until the master brings the nCE line high. Note that if the Address Pointer is 0x7F, it does not increment on the read.

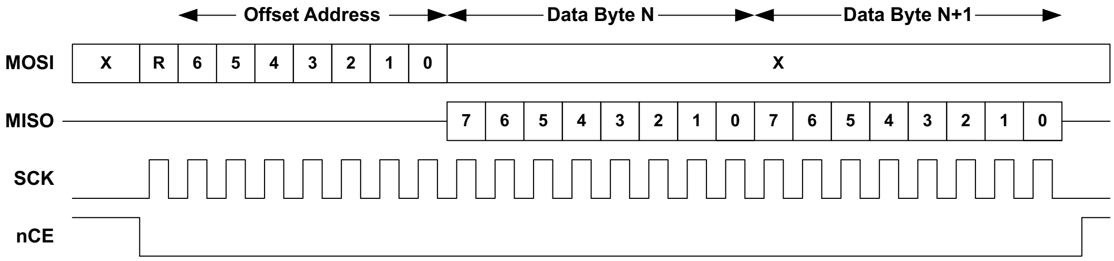


**Figure 33. SPI Read Operation**

### *5.9.3 Configuring 3-wire vs. 4-wire SPI Mode*

The I$^2$C/SPI Slave can operate in either 4-wire SPI mode, where the MISO and MOSI signals are on separate wires, or in 3-wire SPI mode where MISO and MOSI share a wire. This configuration is performed in the Pin Configuration module, and no configuration is necessary in the I$^2$C/SPI Slave itself.

### *5.9.4 SPI Polarity and Phase*

The I$^2$C/SPI Slave supports all combinations of CPOL (clock polarity) and CPHA (data phase) in SPI mode. Figure 34 shows how these two bits affect the interface signal behavior.

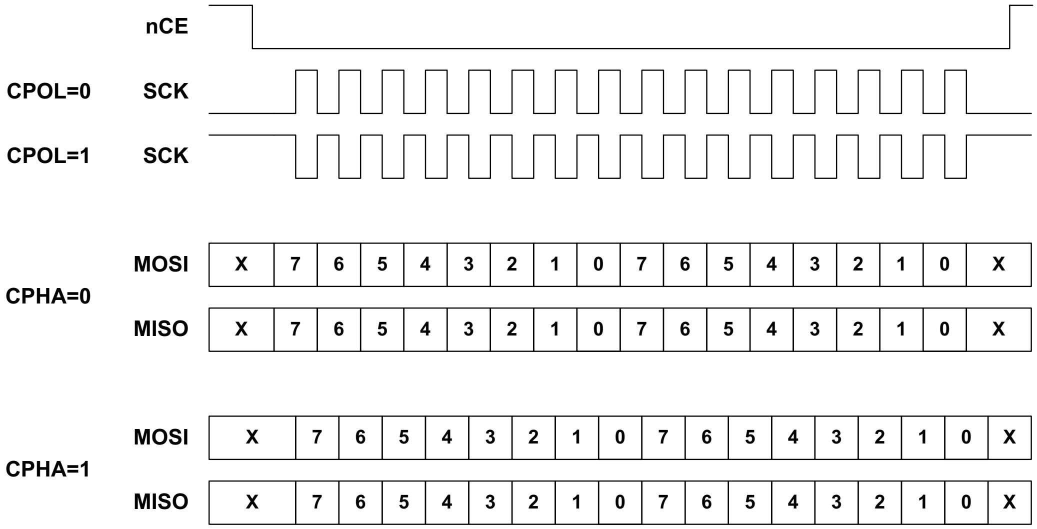

nCE
CPOL=0
SCK
CPOL=1
SCK
CPHA=0
MOSI
X 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 X
MISO
X 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 X
CPHA=1
MOSI
X 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 X
MISO
X 7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0 X

**Figure 34. SPI CPOL and CPHA**

If CPOL is 0, the clock SCK is normally low and positive pulses are generated during transfers. If CPOL is 1, SCK is normally high and negative pulses are generated during transfers.

If CPHA is 0, the data on the MOSI and MISO lines is sampled on the edge corresponding to the first SCK edge after nCE goes low (i.e. the rising edge if CPOL is 0 and the falling edge if CPOL is 1). Data on MISO and MOSI is driven on the opposite edge of SCK.

If CPHA is 1, the data on the MOSI and MISO lines is sampled on the edge corresponding to the second SCK edge after nCE goes low (i.e. the falling edge if CPOL is 0 and the rising edge if CPOL is 1). Data on MISO and MOSI is driven on the opposite edge of SCK.

The $I^2C$/SPI Slave has only a single SPOL bit to control the polarity. If CPOL = CPHA, IOSLAVE_IOSCFG_SPOL must be set to 0. If CPOL ≠ CPHA, SPOL must be set to 1.

## 5.10 Bit Orientation

In both $I^2C$ and SPI modes, the $I^2C$/SPI Slave supports data transmission either LSB first or MSB first as configured by the IOSLAVE_IOSCFG_LSB bit. If LSB is 0, data is transmitted and received MSB first. If LSB is 1, data is transmitted and received LSB first.

## 5.11 Wakeup Using the $I^2C$/SPI Slave

The $I^2C$/SPI Slave can continue to operate even if the MCU’s CPU is in Sleep or Deep Sleep mode. The hardware will enable and disable the $I^2C$/SPI Slave clock and oscillators as necessary. The only consideration in this environment is when the MCU is in a deep sleep mode, such that the HFRC Oscillator is powered down, and a master attempts to access the $I^2C$/SPI Slave. In this case the HFRC Oscillator must be powered up before anything is transferred to or from the internal RAM. This process takes roughly 5-10 us, and is initiated by nCE going low in SPI mode or by the detection of a START in $I^2C$ mode.

For $I^2C$ applications, the time delay is typically not relevant. At the fastest system clock of 1 MHz, the master must transfer 9 bits of address plus 9 bits of offset before any FIFO access can occur, and that is a minimum of 18 us. The clocks will have started prior to that point in every case.

For SPI applications with fast interface clocks (faster than 1 MHz), the master must be programmed to pull nCE low at least 10 us prior to sending the first clock. If a master is unable to control the timing of nCE in this way, then a GPIO interrupt can be configured to wake the MCU prior to initiating any SPI transfers.

There is no delay restriction if the MCU is in normal Sleep mode. In that case the HFRC is not powered down and the $I^2C$/SPI Slave clock will start immediately when nCE goes low.

## 5.12 IOSLAVE Registers

**I2C/SPI Slave**

**INSTANCE 0 BASE ADDRESS:**0x50000000

### *5.12.1 Register Memory Map*

**Table 167: IOSLAVE Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0x50000100 | FIFOPTR | Current FIFO Pointer |
| 0x50000104 | FIFOCFG | FIFO Configuration |
| 0x50000108 | FIFOTHR | FIFO Threshold Configuration |
| 0x5000010C | FUPD | FIFO Update Status |
| 0x50000110 | FIFOCTR | Overall FIFO Counter |
| 0x50000114 | FIFOINC | Overall FIFO Counter Increment |
| 0x50000118 | CFG | I/O Slave Configuration |
| 0x5000011C | PRENC | I/O Slave Interrupt Priority Encode |
| 0x50000120 | IOINTCTL | I/O Interrupt Control |
| 0x50000124 | GENADD | General Address Data |
| 0x50000200 | INTEN | IO Slave Interrupts: Enable |
| 0x50000204 | INTSTAT | IO Slave Interrupts: Status |
| 0x50000208 | INTCLR | IO Slave Interrupts: Clear |
| 0x5000020C | INTSET | IO Slave Interrupts: Set |
| 0x50000210 | REGACCINTEN | Register Access Interrupts: Enable |
| 0x50000214 | REGACCINTSTAT | Register Access Interrupts: Status |
| 0x50000218 | REGACCINTCLR | Register Access Interrupts: Clear |
| 0x5000021C | REGACCINTSET | Register Access Interrupts: Set |

### 5.12.2 IOSLAVE Registers

#### 5.12.2.1 FIFOPTR Register

**Current FIFO Pointer**

**OFFSET:** 0x00000100

**INSTANCE 0 ADDRESS:** 0x50000100

Current FIFO Pointer

**Table 168: FIFOPTR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | FIFOSIZ | | | | | | | | FIFOPTR | | | | | | | |

**Table 169: FIFOPTR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | RSVD | 0x0 | RO | RESERVED |
| 15:8 | FIFOSIZ | 0x0 | RW | The number of bytes currently in the hardware FIFO. |
| 7:0 | FIFOPTR | 0x0 | RW | Current FIFO pointer. |

#### 5.12.2.2 FIFOCFG Register

**FIFO Configuration**

**OFFSET:** 0x00000104

**INSTANCE 0 ADDRESS:** 0x50000104

FIFO Configuration

**Table 170: FIFOCFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | ROBASE | | | | | | RSVD | | | | | | | | RSVD | | FIFOMAX | | | | | | RSVD | | | FIFOBASE | | | | |

**Table 171: FIFOCFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:30 | RSVD | 0x0 | RO | RESERVED |
| 29:24 | ROBASE | 0x20 | RW | Defines the read-only area. The IO Slave read-only area is situated in LRAM at (ROBASE*8) to (FIFOOBASE*8-1) |
| 23:16 | RSVD | 0x0 | RO | RESERVED |
| 15:14 | RSVD | 0x0 | RO | RESERVED |
| 13:8 | FIFOMAX | 0x0 | RW | These bits hold the maximum FIFO address in 8 byte segments. It is also the beginning of the RAM area of the LRAM. Note that no RAM area is configured if FIFOMAX is set to 0x1F. |
| 7:5 | RSVD | 0x0 | RO | RESERVED |
| 4:0 | FIFOBASE | 0x0 | RW | These bits hold the base address of the I/O FIFO in 8 byte segments.The IO Slave FIFO is situated in LRAM at (FIFOBASE*8) to (FIFOMAX*8-1). |

### *5.12.2.3 FIFOTHR Register*

**FIFO Threshold Configuration**

**OFFSET:** 0x00000108

**INSTANCE 0 ADDRESS:** 0x50000108

FIFO Threshold Configuration

**Table 172: FIFOTHR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | FIFOTHR | | | | | | | |

**Table 173: FIFOTHR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED |
| 7:0 | FIFOTHR | 0x0 | RW | FIFO size interrupt threshold. |

### *5.12.2.4 FUPD Register*

**FIFO Update Status**

**OFFSET:** 0x0000010C

**INSTANCE 0 ADDRESS:** 0x5000010C

FIFO Update Status

**Table 174: FUPD Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | IOREAD | FIFOUPD |

**Table 175: FUPD Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:2 | RSVD | 0x0 | RO | RESERVED |
| 1 | IOREAD | 0x0 | RO | This bitfield indicates an IO read is active. |
| 0 | FIFOUPD | 0x0 | RW | This bit indicates that a FIFO update is underway. |

### *5.12.2.5 FIFOCTR Register*

**Overall FIFO Counter**

**OFFSET:** 0x00000110

**INSTANCE 0 ADDRESS:** 0x50000110

Overall FIFO Counter

**Table 176: FIFOCTR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | FIFOCTR | | | | | | | | | |

**Table 177: FIFOCTR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:10 | RSVD | 0x0 | RO | RESERVED |

Table 177: FIFOCTR Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 9:0 | FIFOCTR | 0x0 | RW | Virtual FIFO byte count |

### 5.12.2.6 FIFOINC Register

**Overall FIFO Counter Increment**

**OFFSET:** 0x00000114

**INSTANCE 0 ADDRESS:** 0x50000114

Overall FIFO Counter Increment

Table 178: FIFOINC Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | FIFOINC | | | | | | | | | |

Table 179: FIFOINC Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:10 | RSVD | 0x0 | RO | RESERVED |
| 9:0 | FIFOINC | 0x0 | WO | Increment the Overall FIFO Counter by this value on a write |

### 5.12.2.7 CFG Register

**I/O Slave Configuration**

**OFFSET:** 0x00000118

**INSTANCE 0 ADDRESS:** 0x50000118

I/O Slave Configuration

Table 180: CFG Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IFCEN | RSVD | | | | | | | | | | | I2CADDR | | | | | | | | | | | | RSVD | | | STARTRD | RSVD | LSB | SPOL | IFCSEL |

**Table 181: CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | IFCEN | 0x0 | RW | IOSLAVE interface enable.<br><br>DIS = 0x0 - Disable the IOSLAVE<br>EN = 0x1 - Enable the IOSLAVE |
| 30:20 | RSVD | 0x0 | RO | RESERVED |
| 19:8 | I2CADDR | 0x0 | RW | 7-bit or 10-bit I2C device address.<br>Bit 19 Selects 7-bit/10-bit address: 7BIT = 0x0, 10BIT = 0x1<br>Bit 18:16 Upper 3 bits for 10-bit address; not used for 7-bit address<br>Bit 15:9 7-bit address or lower 7 bits of 10-bit address<br>Bit 8 Not used for either address mode |
| 7:5 | RSVD | 0x0 | RO | RESERVED |
| 4 | STARTRD | 0x0 | RW | This bit holds the cycle to initiate an I/O RAM read.<br><br>LATE = 0x0 - Initiate I/O RAM read late in each transferred byte.<br>EARLY = 0x1 - Initiate I/O RAM read early in each transferred byte. |
| 3 | RSVD | 0x0 | RO | RESERVED |
| 2 | LSB | 0x0 | RW | This bit selects the transfer bit ordering.<br><br>MSB_FIRST = 0x0 - Data is assumed to be sent and received with MSB first.<br>LSB_FIRST = 0x1 - Data is assumed to be sent and received with LSB first. |
| 1 | SPOL | 0x0 | RW | This bit selects SPI polarity.<br><br>SPI_MODES_0_3 = 0x0 - Polarity 0, handles SPI modes 0 and 3.<br>SPI_MODES_1_2 = 0x1 - Polarity 1, handles SPI modes 1 and 2. |
| 0 | IFCSEL | 0x0 | RW | This bit selects the I/O interface.<br><br>I2C = 0x0 - Selects I2C interface for the IO Slave.<br>SPI = 0x1 - Selects SPI interface for the IO Slave. |

### 5.12.2.8 PRENC Register

**I/O Slave Interrupt Priority Encode**

**OFFSET:** 0x0000011C

**INSTANCE 0 ADDRESS:** 0x5000011C

I/O Slave Interrupt Priority Encode

**Table 182: PRENC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | PRENC | | | | |

**Table 183: PRENC Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:5 | RSVD | 0x0 | RO | RESERVED |
| 4:0 | PRENC | 0x0 | RO | These bits hold the priority encode of the REGACC interrupts. |

### *5.12.2.9 IOINTCTL Register*

**I/O Interrupt Control**

**OFFSET:** 0x00000120

**INSTANCE 0 ADDRESS:** 0x50000120

I/O Interrupt Control

**Table 184: IOINTCTL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IOINTSET | | | | | | | | RSVD | | | | | | | IOINTCLR | IOINT | | | | | | | | IOINTEN | | | | | | | |

**Table 185: IOINTCTL Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:24 | IOINTSET | 0x0 | WO | These bits set the IOINT interrupts when written with a 1. |
| 23:17 | RSVD | 0x0 | RO | RESERVED |
| 16 | IOINTCLR | 0x0 | WO | This bit clears all of the IOINT interrupts when written with a 1. |
| 15:8 | IOINT | 0x0 | RO | These bits read the IOINT interrupts. |
| 7:0 | IOINTEN | 0x0 | RO | These bits setread the IOINT interrupt enables. |

### *5.12.2.10 GENADD Register*

**General Address Data**

**OFFSET:** 0x00000124

**INSTANCE 0 ADDRESS:** 0x50000124

General Address Data

**Table 186: GENADD Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | GADATA | | | | | | | |

**Table 187: GENADD Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED |
| 7:0 | GADATA | 0x0 | RO | The data supplied on the last General Address reference. |

### *5.12.2.11 INTEN Register*

**IO Slave Interrupts: Enable**

**OFFSET:** 0x00000200

**INSTANCE 0 ADDRESS:** 0x50000200

Set bits in this register to allow this module to generate the corresponding interrupt.

**Table 188: INTEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | IOINTW | GENAD | FRDERR | FUNDFL | FOVFL | FSIZE |

**Table 189: INTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:6 | RSVD | 0x0 | RO | RESERVED |
| 5 | IOINTW | 0x0 | RW | Interrupt Write interrupt. |

Table 189: INTEN Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 4 | GENAD | 0x0 | RW | I2C General Address interrupt. |
| 3 | FRDERR | 0x0 | RW | FIFO Read Error interrupt. |
| 2 | FUNDFL | 0x0 | RW | FIFO Underflow interrupt. |
| 1 | FOVFL | 0x0 | RW | FIFO Overflow interrupt. |
| 0 | FSIZE | 0x0 | RW | FIFO Size interrupt. |

### *5.12.2.12 INTSTAT Register*

**IO Slave Interrupts: Status**

**OFFSET:** 0x00000204

**INSTANCE 0 ADDRESS:** 0x50000204

Read bits from this register to discover the cause of a recent interrupt.

Table 190: INTSTAT Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | IOINTW | GENAD | FRDERR | FUNDFL | FOVFL | FSIZE |

Table 191: INTSTAT Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:6 | RSVD | 0x0 | RO | RESERVED |
| 5 | IOINTW | 0x0 | RW | Interrupt Write interrupt. |
| 4 | GENAD | 0x0 | RW | I2C General Address interrupt. |
| 3 | FRDERR | 0x0 | RW | FIFO Read Error interrupt. |
| 2 | FUNDFL | 0x0 | RW | FIFO Underflow interrupt. |
| 1 | FOVFL | 0x0 | RW | FIFO Overflow interrupt. |
| 0 | FSIZE | 0x0 | RW | FIFO Size interrupt. |

### *5.12.2.13 INTCLR Register*

**IO Slave Interrupts: Clear**

**OFFSET:** 0x00000208

**INSTANCE 0 ADDRESS:** 0x50000208

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

**Table 192: INTCLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | IOINTW | GENAD | FRDERR | FUNDFL | FOVFL | FSIZE |

**Table 193: INTCLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:6 | RSVD | 0x0 | RO | RESERVED |
| 5 | IOINTW | 0x0 | RW | Interrupt Write interrupt. |
| 4 | GENAD | 0x0 | RW | I2C General Address interrupt. |
| 3 | FRDERR | 0x0 | RW | FIFO Read Error interrupt. |
| 2 | FUNDFL | 0x0 | RW | FIFO Underflow interrupt. |
| 1 | FOVFL | 0x0 | RW | FIFO Overflow interrupt. |
| 0 | FSIZE | 0x0 | RW | FIFO Size interrupt. |

### *5.12.2.14 INTSET Register*

**IO Slave Interrupts: Set**

**OFFSET:** 0x0000020C

**INSTANCE 0 ADDRESS:** 0x5000020C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

**Table 194: INTSET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | IOINTW | GENAD | FRDERR | FUNDFL | FOVFL | FSIZE |

**Table 195: INTSET Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:6 | RSVD | 0x0 | RO | RESERVED |
| 5 | IOINTW | 0x0 | RW | Interrupt Write interrupt. |
| 4 | GENAD | 0x0 | RW | I2C General Address interrupt. |
| 3 | FRDERR | 0x0 | RW | FIFO Read Error interrupt. |
| 2 | FUNDFL | 0x0 | RW | FIFO Underflow interrupt. |
| 1 | FOVFL | 0x0 | RW | FIFO Overflow interrupt. |
| 0 | FSIZE | 0x0 | RW | FIFO Size interrupt. |

### 5.12.2.15 REGACCINTEN Register

**Register Access Interrupts: Enable**

**OFFSET:** 0x00000210

**INSTANCE 0 ADDRESS:** 0x50000210

Set bits in this register to allow this module to generate the corresponding interrupt.

**Table 196: REGACCINTEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| REGACC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 197: REGACCINTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | REGACC | 0x0 | RW | Register access interrupts. |

### 5.12.2.16 REGACCINTSTAT Register

**Register Access Interrupts: Status**

**OFFSET:** 0x00000214

**INSTANCE 0 ADDRESS:** 0x50000214

Read bits from this register to discover the cause of a recent interrupt.

**Table 198: REGACCINTSTAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| REGACC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 199: REGACCINTSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | REGACC | 0x0 | RW | Register access interrupts. |

### 5.12.2.17 REGACCINTCLR Register

**Register Access Interrupts: Clear**

**OFFSET:** 0x00000218

**INSTANCE 0 ADDRESS:** 0x50000218

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

**Table 200: REGACCINTCLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| REGACC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 201: REGACCINTCLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | REGACC | 0x0 | RW | Register access interrupts. |

### 5.12.2.18 REGACCINTSET Register

**Register Access Interrupts: Set**

**OFFSET:** 0x0000021C

**INSTANCE 0 ADDRESS:** 0x5000021C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

**Table 202: REGACCINTSET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| REGACC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 203: REGACCINTSET Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | REGACC | 0x0 | RW | Register access interrupts. |

## 5.13 Host Side Address Space and Register

### *5.13.1 Host Address Space and Registers*

The Host of the I/O interface can access 128 bytes in the I$^2$C/SPI Slave in either I$^2$C or SPI mode. Offsets 0x00 to 0x77 may be directly mapped to the Direct RAM Area. The remaining eight offset locations access hardware functions within the I$^2$C/SPI Slave. The R/W indicator refers to accesses from the Host.

#### *5.13.1.1 HOST_IER Register*

**Host Interrupt Enable**

**OFFSET:** 0x78

This register enables the FIFO read interrupts.

**Table 204: HOST_IER Register**

| 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|
| FUNDFLEN | RDERREN | SWINT5EN | SWINT4EN | SWINT3EN | SWINT2EN | SWINT1EN | SWINT0EN |

**Table 205: HOST_IER Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 7 | FUNDFLEN | 0x0 | RW | If 1, enable an interrupt that triggers when the FIFO underflows |
| 6 | RDERREN | 0x0 | RW | If 1, enable the interrupt which occurs when the Host attempts to access the FIFO when read access is locked |
| 5 | SWINT5EN | 0x0 | RW | If 1, enable software interrupt 5 |
| 4 | SWINT4EN | 0x0 | RW | If 1, enable software interrupt 4 |
| 3 | SWINT3EN | 0x0 | RW | If 1, enable software interrupt 3 |
| 2 | SWINT2EN | 0x0 | RW | If 1, enable software interrupt 2 |
| 1 | SWINT1EN | 0x0 | RW | If 1, enable software interrupt 1 |
| 0 | SWINT0EN | 0x0 | RW | If 1, enable software interrupt 0 |

### *5.13.1.2 HOST_ISR Register*

**Host Interrupt Status Register**

**OFFSET:** 0x79

The host uses this register to read interrupt status.

**Table 206: HOST_ISR Register**

| 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|
| FUNDFLSTAT | RDERRSTAT | SWINT5STAT | SWINT4STAT | SWINT3STAT | SWINT2STAT | SWINT1STAT | SWINT0STAT |

**Table 207: HOST_ISR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 7 | FUNDFLSTAT | 0x0 | RO | This bit is set by writing a 1 to bit 31 of the IOINTCTL Register, or if the Host attempts a FIFO read when FIFOCTR is 0. |
| 6 | RDERRSTAT | 0x0 | RO | This bit is set by writing a 1 to bit 30 of the IOINTCTL Register, or if the Host attempts a FIFO read when the FIFOUPD bit is a 1. |
| 5 | SWINT5STAT | 0x0 | RO | This bit is set by writing a 1 to bit 29 of the IOINTCTL Register. |
| 4 | SWINT4STAT | 0x0 | RO | This bit is set by writing a 1 to bit 28 of the IOINTCTL Register. |
| 3 | SWINT3STAT | 0x0 | RO | This bit is set by writing a 1 to bit 27 of the IOINTCTL Register. |
| 2 | SWINT2STAT | 0x0 | RO | This bit is set by writing a 1 to bit 26 of the IOINTCTL Register. |
| 1 | SWINT1STAT | 0x0 | RO | This bit is set by writing a 1 to bit 25 of the IOINTCTL Register. |
| 0 | SWINT0STAT | 0x0 | RO | This bit is set by writing a 1 to bit 24 of the IOINTCTL Register. |

NOTE
All bits are cleared by a write to the IOINTCLR bit of the IOINTCTL Register.

### *5.13.1.3 HOST_WCR Register*

**Host Interrupt Write-to-Clear Register**

**OFFSET:** 0x7A

Write a 1 to a bit in this register to clear a pending interrupt.

**Table 208: HOST_WCR Register**

| 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|
| FUNDFLWC | RDERRWC | SWINT5WC | SWINT4WC | SWINT3WC | SWINT2WC | SWINT1WC | SWINT0WC |

**Table 209: HOST_WCR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 7 | FUNDFLWC | 0x0 | WO | Writing a 1 to this bit will clear the pending interrupt status bit FUNDFLSTAT |
| 6 | RDERRWC | 0x0 | WO | Writing a 1 to this bit will clear the pending interrupt status bit RDERRSTAT |
| 5 | SWINT5WC | 0x0 | WO | Writing a 1 to this bit will clear the pending interrupt status bit SWINT5STAT |
| 4 | SWINT4WC | 0x0 | WO | Writing a 1 to this bit will clear the pending interrupt status bit SWINT4STAT |
| 3 | SWINT3WC | 0x0 | WO | Writing a 1 to this bit will clear the pending interrupt status bit SWINT3STAT |
| 2 | SWINT2WC | 0x0 | WO | Writing a 1 to this bit will clear the pending interrupt status bit SWINT2STAT |
| 1 | SWINT1WC | 0x0 | WO | Writing a 1 to this bit will clear the pending interrupt status bit SWINT1STAT |
| 0 | SWINT0WC | 0x0 | WO | Writing a 1 to this bit will clear the pending interrupt status bit SWINT0STAT |

### 5.13.1.4 HOST_WCS Register

**Host Interrupt Write-to-Set Register**

**OFFSET:** 0x7B

Write a 1 to a bit in this register to set the status bit of a pending interrupt.

**Table 210: HOST_WCS Register**

| 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|
| FUNDFLWS | RDERRWS | SWINT5WS | SWINT4WS | SWINT3WS | SWINT2WS | SWINT1WS | SWINT0WS |

**Table 211: HOST_WCS Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 7 | FUNDFLWS | 0x0 | WO | Writing a 1 to this bit will set the pending interrupt status bit FUNDFLSTAT |
| 6 | RDERRWS | 0x0 | WO | Writing a 1 to this bit will set the pending interrupt status bit RDERRSTAT |
| 5 | SWINT5WS | 0x0 | WO | Writing a 1 to this bit will set the pending interrupt status bit SWINT5STAT |
| 4 | SWINT4WS | 0x0 | WO | Writing a 1 to this bit will set the pending interrupt status bit SWINT4STAT |
| 3 | SWINT3WS | 0x0 | WO | Writing a 1 to this bit will set the pending interrupt status bit SWINT3STAT |
| 2 | SWINT2WS | 0x0 | WO | Writing a 1 to this bit will set the pending interrupt status bit SWINT2STAT |
| 1 | SWINT1WS | 0x0 | WO | Writing a 1 to this bit will set the pending interrupt status bit SWINT1STAT |
| 0 | SWINT0WS | 0x0 | WO | Writing a 1 to this bit will set the pending interrupt status bit SWINT0STAT |

### 5.13.1.5 FIFOCTRLO Register

**FIFOCTR Low Byte**

**OFFSET:** 0x7C

This register allows the host to read the lower eight bits of the FIFOCTR register.

**Table 212: FIFOCTRLO Register**

| 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|
| FIFOCTRLO | | | | | | | |

**Table 213: FIFOCTRLO Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 7:0 | FIFOCTRLO | 0x0 | RO | Reads the lower eight bits of FIFOCTR |

### 5.13.1.6 FIFOCTRUP Register

**FIFOCTR Upper Byte**

**OFFSET:** 0x7D

This register allows the host to read the upper two bits of the FIFOCTR register.

**Table 214: FIFOCTRUP Register**

| 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|
| RSVD | | | | | | FIFOCTRUP | |

**Table 215: FIFOCTRUP Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 1:0 | FIFOCTRUP | 0x0 | RO | Reads the upper two bits of FIFOCTR |

### 5.13.1.7 FIFO Register

**FIFO Read Data**

**OFFSET:** 0x7F

Read this register for FIFO data.

**Table 216: FIFO Register**

<table>
<tr><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="8">FIFO</td></tr>
</table>

**Table 217: FIFO Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 7:0 | FIFO | 0x0 | RO | Reads the top byte of the FIFO |

# 6. GPIO and Pad Configuration Module

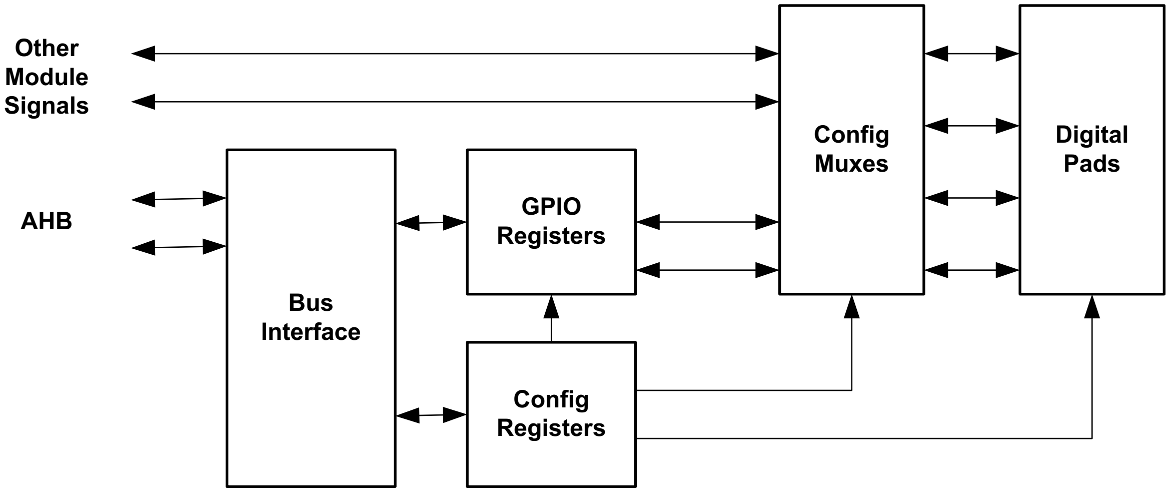


**Figure 35. Block diagram for the General Purpose I/O (GPIO) Module**

## 6.1 Functional Overview

The General Purpose I/O and Pad Configuration (GPIO) Module, shown in Figure 35, controls connections to up to 50 digital/analog pads. Each pad may be connected to a variety of module interface signals, with all pad input and output selection and control managed by the GPIO module. In addition, any pad may function as a general purpose input and/or output pad which may be configured for a variety of external functions. Each GPIO may be configured to generate an interrupt when a transition occurs on the input.

## 6.2 Pad Configuration Functions

The GPIO_PADREGy (y = A to M) registers are used to control the function of each pad. Note that the GPIO_PADKEY Register must be set to the value 0x73 in order to write the PADREGn registers.

NOTE

Once the PADKEY is written, it should be explicitly cleared (with a non-key value) after GPIO configuration register updates are complete.

The GPIO_PADREGy_PADnFNCSEL (n = 0 to 49) field selects one of up to eight signals to be used for each pad, as shown in Table 218. Functions are grouped by module, with the color coding shown in Table 219. This table also defines the pad type for each configuration. The Special Pad Types are defined in Table 220. Note that the CSP package only supports pads 0 through 21, 29, 31, 35, 36 and 41, which are shaded in orange in the Pad column of Table 218.

The REG_GPIO_PADREGy_PADnSTRNG bit controls the drive strength of the pad. When this bit is set, the pad is configured for high drive strength.

For all pads except for pad 20, GPIO_PADREGy_PADnPULL bit enables a weak pull-up on the pad when set to one. For pad 20, the GPIO_PADREGy_PAD20PULL bit enables a weak pull-down on the pad when set to one. The GPIO_PADREGy_PADnINPEN bit must be set to enable the pad input, and should be left clear whenever the pad is not used in order to eliminate any leakage current in the pad.

Pads with a (*) (pads 3 and 4) have selectable high side power switch transistors to provide ~1 Ω switches to VDDH. Pads with (**) (pad 11) have selectable low side power switch transistors to provide ~1 Ω switches to VSS. A high side power switch is enabled by setting the REG_GPIO_PADREGy_PAD3PWRUP or REG_GPIO_PADREGy_PAD4PWRUP bit, and a low side switch is enabled by setting the REG_GPIO_PADREGy_PAD11PWRDN bit. Once enabled, the switches operate in parallel with the normal pad function.

Pads 5, 6, 8 and 9 include optional pull-up resistors for use in $I^2C$ mode, to eliminate the need for external resistors. If the pull-up is enabled by the PADnPULL bit, the GPIO_PADREGy_PADnRSEL field selects the size of the pull-up resistor as shown in Table 221.

**Table 218: Apollo Pad Function Mapping**

| Pad | PADnFNCSEL | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | SLSCL [I] | SLSCK [I] | UARTTX [O] | GPIO0 | M0SCK_LB | M1SCK_LB | M0SCL_LB | M1SCL_LB |
| 1 | SLSDA [S] | SLMISO [O] | UARTRX [I] | GPIO1 | M0MISO_LB | M1MISO_LB | M0SDA_LB | M1SDA_LB |
| 2 | SLWIR3 [S] | SLMOSI [I] | CLKOUT | GPIO2 | M0MOSI_LB | M1MOSI_LB | M0WIR3_LB | M1WIR3_LB |
| 3* | TRIG0 [I] | SLnCE [O] | M1nCE4 | GPIO3 | M0nCE_LB | M1nCE_LB | | |
| 4* | TRIG1 [I] | SLINT [O] | M0nCE5 | GPIO4 | SLINTGP_LB | SWO [O] | CLKOUT | |
| 5 | M0SCL [S] | M0SCK[O] | UARTS [O] | GPIO5 | M0SCK_LB | | M0SCL_LB | |
| 6 | M0SDA [S] | M0MISO [I] | UACTS [I] | GPIO6 | SLMISO_LB | | SLSDA_LB | |
| 7 | M0WIR3 [S] | M0MOSI [O] | CLKOUT | GPIO7 | | | SLWIR3_LB | |
| 8 | M1SCL [S] | M1SCK [O] | M0nCE4 | GPIO8 | | M1SCK_LB | | M1SCL_LB |
| 9 | M1SDA [S] | M1MISO [I] | M0nCE5 | GPIO9 | | SLMISO_LB | | SLSDA_LB |
| 10 | M1WIR3 [S] | M1MOSI [O] | M0nCE6 | GPIO10 | | | | SLWIR3_LB |
| 11** | RESERVED | M0nCE0 | CLKOUT | GPIO11 | | | | |
| 12 | ADC0 [A] | M1nCE0 | TCTA0 | GPIO12 | | | | |
| 13 | ADC1 [A] | M1nCE1 | TCTB0 | GPIO13 | | | SWO [O] | |
| 14 | ADC2 [A] | M1nCE2 | UARTTX [O] | GPIO14 | | | | |
| 15 | ADC3 [A] | M1nCE3 | UARTRX [I] | GPIO15 | | | | |
| 16 | ADCREF [A] | M0nCE4 | TRIG2 [I] | GPIO16 | | | | |
| 17 | CMPAD0 [A] | M0nCE1 | TRIG3 [I] | GPIO17 | | | | |
| 18 | CMPAD1 [A] | M0nCE2 | TCTA1 | GPIO18 | | | | |
| 19 | CMPRF0 [A] | M0nCE3 | TCTB1 | GPIO19 | | | | |
| 20 | SWDCK [I] | M1nCE5 | TCTA2 | GPIO20 | | | | |
| 21 | SWDIO [S] | M1nCE6 | TCTB2 | GPIO21 | | | | |
| 22 | UARTTX [O] | M1nCE7 | TCTA3 | GPIO22 | | | | |
| 23 | UARTRX [I] | M0nCE0 | TCTB3 | GPIO23 | | | | |
| 24 | | M0nCE1 | CLKOUT | GPIO24 | | | | |
| 25 | | M0nCE2 | TCTA0 | GPIO25 | | | | |
| 26 | | M0nCE3 | TCTB0 | GPIO26 | | | | |
| 27 | | M1nCE4 | TCTA1 | GPIO27 | | | | |
| 28 | | M1nCE5 | TCTB1 | GPIO28 | | | | |
| 29 | ADC4 [A] | M1nCE6 | TCTA2 | GPIO29 | | | | |
| 30 | ADC5 [A] | M1nCE7 | TCTB2 | GPIO30 | | | | |
| 31 | ADC6 [A] | M0nCE4 | TCTA3 | GPIO31 | | | | |
| 32 | ADC7 [A] | M0nCE5 | TCTB3 | GPIO32 | | | | |
| 33 | CMPRF1 [A] | M0nCE6 | | GPIO33 | | | | |
| 34 | CMPRF2 [A] | M0nCE7 | | GPIO34 | | | | |
| 35 | | M1nCE0 | UARTTX [O] | GPIO35 | | | | |
| 36 | | M1nCE1 | UARTRX [I] | GPIO36 | | | | |
| 37 | TRIG0 [I] | M1nCE2 | UARTS [O] | GPIO37 | | | | |
| 38 | TRIG1 [I] | M1nCE3 | UACTS [I] | GPIO38 | | | | |
| 39 | TRIG2 [I] | UARTTX [O] | CLKOUT | GPIO39 | | | | |
| 40 | TRIG3 [I] | UARTRX [I] | | GPIO40 | | | | |
| 41 | TRIG4 [I] | | SWO [O] | GPIO41 | | | | |
| 42 | TRIG5 [I] | M0nCE0 | TCTA0 | GPIO42 | | | | |
| 43 | TRIG6 [I] | M0nCE1 | TCTB0 | GPIO43 | | | | |
| 44 | TRIG7 [I] | M0nCE2 | TCTA1 | GPIO44 | | | | |
| 45 | | M0nCE3 | TCTB1 | GPIO45 | | | | |
| 46 | | M0nCE4 | TCTA2 | GPIO46 | | | | |
| 47 | | M0nCE5 | TCTB2 | GPIO47 | | | | |
| 48 | | M0nCE6 | TCTA3 | GPIO48 | | | | |
| 49 | | M0nCE7 | TCTB3 | GPIO49 | | | | |

### Table 219: Pad Function Color and Symbol Code

| Color/ Symbol | Function | Pad Type |
|---|---|---|
| | | |
| 🟧 | Various | Supported on CSP package |
| 🟦 | ADC Signals | Analog or Input, as indicated by [A] or [I] respectively |
| 🟩 | I$^2$C/SPI Slave Signals | Input, Special or Push-pull output, as indicated by [I], [S] or [O] respectively, |
| 🟥 | I$^2$C/SPI Master 0 Signals | Input, Special or Push-pull output, as indicated by [I], [S] or [O] respectively |
| 🟨 | I$^2$C/SPI Master 1 Signals | Input, Special or Push-pull output, as indicated by [I], [S] or [O] respectively |
| 🟨 | GPIO Signals | Controlled by GPIO Configuration |
| 🟪 | Counter/Timer Signals | Controlled by CTIMER Configuration |
| 🟩 | UART Signals | Input or Push-pull output, as indicated by [I] or [O] respectively |
| 🟩 | CLKOUT Signals | Push-pull Output |
| 🟥 | Loopback Connections | Tri-state |
| 🟦 | Miscellaneous Signals | Input Special or Push-pull output, as indicated by [I], [S] or [O] respectively |
| * | High-side power switch | Pads with a (*) (pads 3 and 4) have selectable high side power switch transistors to provide ~1 Ω switches to VDDH. |
| ** | Low-side power switch | Pads with a (**) (pad 11) have selectable low side power switch transistors to provide ~1 Ω switches to VSS. |

### Table 220: Special Pad Types

| Pad | PADnFNCSEL | Name | Pad Type |
|---|---|---|---|
| 1 | 0 | SLSDA | Bidirectional Open Drain |
| 2 | 0 | SLWIR3 | Bidirectional Tri-state |
| 5 | 0 | M0SCL | Open Drain |
| 6 | 0 | M0SDA | Bidirectional Open Drain |
| 7 | 0 | M0WIR3 | Bidirectional Tri-state |
| 8 | 0 | M1SCL | Open Drain |
| 9 | 0 | M1SDA | Bidirectional Open Drain |
| 10 | 0 | M1WIR3 | Bidirectional Tri-state |
| 21 | 0 | SWDIO | Bidirectional Tri-state |

**Table 221: $I^2C$ Pullup Resistor Selection**

| RSEL[1:0] | Pullup Resistor |
|---|---|
| 00 | 1.5 kΩ |
| 01 | 6 kΩ |
| 10 | 12 kΩ |
| 11 | 24 kΩ |

## 6.3 General Purpose I/O (GPIO) Functions

For each pad, if the PADnFNCSEL field is set to 0x3 the pad is connected to the corresponding GPIO signal. This section describes the configuration functions specific to GPIO pads.

### *6.3.1 Configuring the GPIO Functions*

Each GPIO must be configured in the GPIO_CFGy (y = A to G) Registers as an input and/or output before using. Note that the PADKEY Register must be set to the value 0x73 in order to write the GPIO_CFGy Registers. Each output may be push-pull, open drain, disabled, or tristated as selected by the GPIO_CFGy_GPIOnOUTCFG field. If the output is configured as push-pull, the pad will be driven with the corresponding bit in the GPIO_WTy (y = A or B) Register. If the output is configured as open drain, the pad will be pulled low if the corresponding bit in the WTy Register is a 0, and will be floating if the corresponding bit in the WTy Register is a 1. If the output is configured as tri-state, the pad will be driven with the corresponding bit in the WTy Register if the corresponding bit in the GPIO_ENy Register is a 1. If the bit in ENy is a 0, the output will be floating.

### *6.3.2 Reading from a GPIO Pad*

All GPIO inputs are readable at all times, even if the pad is not configured as a GPIO. The current values of pads 0 to 31 are read in the GPIO_RDA Register, and the current values of pads 32 to 49 are read in the GPIO_RDB Register. If the GPIO_CFGy_GPIOnINCFG bit is set for a GPIO, it will always read as zero.

### *6.3.3 Writing to a GPIO Pad*

The GPIO pad outputs are controlled by the GPIO_WTA/B Registers and the GPIO_ENA/B Registers. Each of these registers may be directly written and read. Because each GPIO is often an independent function, the capability also exists to set or clear one or more bits without having to perform a read-modify-write operation. If the GPIO_WTSA/B Register is written, the corresponding bit in WTA/B will be set if the write data is 1, otherwise the WTA/B bit will not be changed. If the GPIO_WTCA/B Register is written, the corresponding bit in WTA/B will be cleared if the write data is 1, otherwise the WTA/B bit will not be changed.

If a GPIO pad is configured for tri-state output mode, the ENA/B Register controls the enabling of each bit. These registers may be directly written, and individual bits may be set or cleared by writing the ENSA/B or ENCA/B Registers with a 1 in the desired bit position.

### *6.3.4 GPIO Interrupts*

Each GPIO pad can be configured to generate an interrupt on a high-to-low transition or a low-to-high transition, as needed by setting the GPIO_CFGy_GPIOnINTD  bit (y = A to G, n = 0 to 49). This interrupt will be generated even if the pad is not configured as a GPIO in the Pad Configuration logic.

Each interrupt is enabled, disabled, cleared or set with a standard set of interrupt registers GPIO_INT0EN, GPIO_INT0STAT, GPIO_INT0CLR and GPIO_INT0SET for GPIO pads 0 to 31, and Registers GPIO_INT1EN, GPIO_INT1STAT, GPIO_INT1CLR and GPIO_INT1SET for GPIO pads 32 to 49.

Note that these interrupts get mapped to different IRQs and hence respective interrupts need to be enabled in NVIC and serviced accordingly.

## 6.4 Pad Connection Summary

Figure 36 shows the detailed implementation of each pad. Each element will be described in detail.

### *6.4.1 Output Selection*

There is a multiplexer (4 or 8 input depending upon the pad) which selects the module signal to be driven to the output based on GPIO_PADREGy_PADnFNCSEL (y = A to M (n = 0 to 49)) field. This implements the multiplexing shown in Table 218 for output pads. For all pads, a PADnFNCSEL value of 0x3 selects the value in the corresponding GPIO_WTy register bit.

### *6.4.2 Output Control*

The pad driver for each pad has a data input and an output enable input. Each of these controls is selected from among several alternatives based on the OUTDATSEL and OUTENSEL signals which are controlled by the selection of the output type as shown in Table 220 and Table 221.

OUTDATSEL normally selects the data from the output multiplexer, but if the pad is configured as Open Drain the data input is selected to be low.

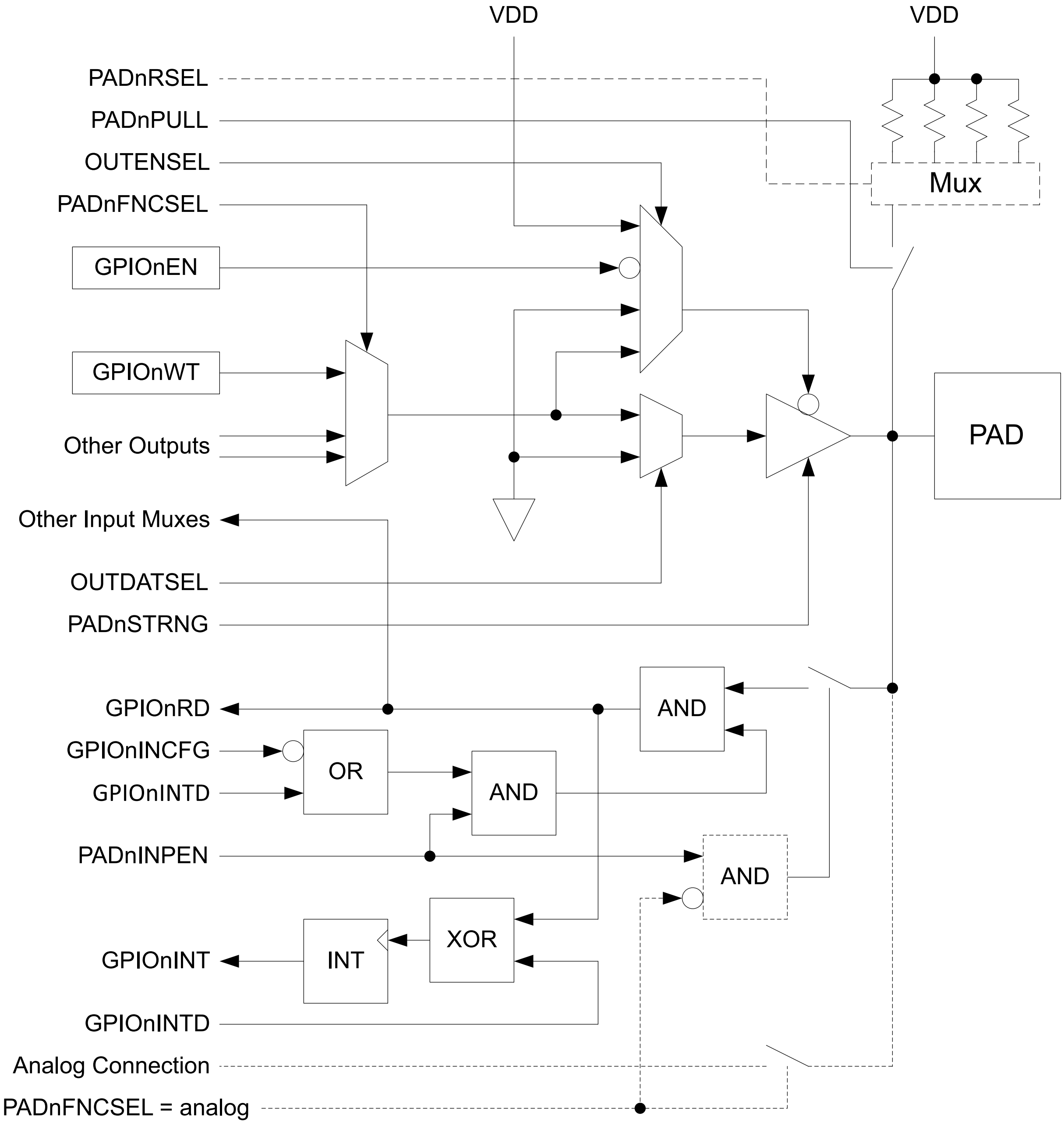


**Figure 36. Pad Connection Details**

OUTENSEL normally selects a ground signal to keep the pad driver enabled. If the pad is configured to be Open Drain, the pad enable is driven with the data from the output multiplexer. If the pad is configured as a GPIO (PADnFNCSEL = 0x3) and the GPIO drive type is tri-state (GPIOnOUTCFG = 0x3), the pad enable is driven with the inverse of the corresponding GPIO bit in the GPIO_ENx register. If the pad is not configured as an output, the pad enable is forced high to turn the driver off.

The drive strength of each pad driver is configured as described in Section 6.2 on page 140.

### 6.4.3 Input Control

The input circuitry of the pad may be disabled by clearing the PADnINPEN bit. This configuration should always be set if the pad input is not being used, as it prevents unnecessary current consumption if the pad voltage happens to float to a level between VDD and Ground. If PADnINPEN is 0, the pad will always read as a 0.

If PADnINPEN is set, the pad input then goes to two places. It is driven to the selected module signal as selected in Table 218. In addition, the pad input can always be read from the GPIO_RDx register unless the pad is configured as a GPIO (PADnFNCSEL = 0x3) and GPIOnINCFG is high, which will force the GPIO_RD input to be a zero. The ability to always read the pad value is very useful in some diagnostic cases.

The pad input is always sent to the GPIO interrupt logic, and a pad transition in the direction selected by GPIOnINTD will set the GPIOn_INT flip-flop. Note that this interrupt will be set even if the pad is not configured as a GPIO, which may be useful in detecting functions. As an example, this could be used to generate an interrupt when the I$^2$C/SPI Slave nCE signal is driven low by the Interface Host.

### 6.4.4 Pull-up Control

If PADnPULL is high, a pull-up resistor is connected between the pad and VDDH, except for pad 20, where PADnPULL connects the resistor to VSS rather than VDDH.

The four pads which can be I$^2$C/SPI Master output drivers (5, 6, 8, and 9) contain the additional circuitry required for this functionality. In this case one of four different pull-up resistors are selected among options for the PADnRSEL field.

### 6.4.5 Analog Pad Configuration

Pads which may have analog connections (11-19 and 29-34) include the circuitry shown with the dotted lines of Figure 36. If the pad is configured in analog mode (PADnFNCSEL is 0x0), the pad is connected directly to the particular analog module signal. In addition, OUTENSEL is forced high to disable the pad output, and the input of the pad is disabled independent of the value of PADnINPEN.

## 6.5 Module-specific Pad Configuration

The following sections describe in detail how to configure the pads for each module function.

### 6.5.1 Implementing IO Master Connections

The two IO Master modules must be correctly connected to the appropriate pads in order to operate.

#### 6.5.1.1 IO Master 0 I$^2$C Connection

I$^2$C mode of IO Master 0 uses pad 5 as SCL and pad 6 as SDA. This mode is configured by setting the PADnFNCSEL fields as shown in Table 222. The PAD5INPEN and PAD6INPEN bits must be set. If the internal I$^2$C pullup resistors are to be used, PAD5PULL and PAD6PULL should be set, and the PAD5RSEL and PAD6RSEL fields should be set to select the desired pullup resistor size as shown in Table 221. If external pullup resistors are used, PAD5PULL and PAD6PULL should be cleared.

**Table 222: IO Master 0 I$^2$C Configuration**

| Field | Value |
|---|---|
| PAD5FNCSEL | 0 |
| PAD6FNCSEL | 0 |

#### 6.5.1.2 IO Master 1 $I^2C$ Connection

$I^2C$ mode of IO Master 1 uses pad 8 as SCL and pad 9 as SDA. This mode is configured by setting the PADnFNCSEL fields as shown in Table 223. The PAD8INPEN and PAD9INPEN bits must be set. If the internal $I^2C$ pullup resistors are to be used, PAD8PULL and PAD9PULL should be set, and the PAD8RSEL and PAD9RSEL fields should be set to select the desired pullup resistor size as shown in Table 221. If external pullup resistors are used, PAD8PULL and PAD9PULL should be cleared.

**Table 223: IO Master 1 $I^2C$ Configuration**

| Field | Value |
|---|---|
| PAD8FNCSEL | 0 |
| PAD9FNCSEL | 0 |

#### 6.5.1.3 IO Master 0 4-wire SPI Connection

Four-wire SPI mode of IO Master 0 uses pad 5 as SCK, pad 6 as MISO and pad 7 as MOSI. This mode is configured by setting the PADnFNCSEL fields as shown in Table 224. The PAD5INPENand PAD6INPEN bits must be set. PAD5PULL, PAD6PULL and PAD7PULL should be cleared. A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 225. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

**Table 224: IO Master 0 4-wire SPI Configuration**

| Field | Value |
|---|---|
| PAD5FNCSEL | 1 |
| PAD6FNCSEL | 1 |
| PAD7FNCSEL | 1 |

**Table 225: IO Master 0 4-wire SPI nCE Configuration**

| Field | Value | nCE Selection | Pad Used |
|---|---|---|---|
| PAD11FNCSEL | 1 | 0 | 11 |
| PAD23FNCSEL | 1 | 0 | 23 |
| PAD42FNCSEL | 1 | 0 | 42 |
| PAD17FNCSEL | 1 | 1 | 17 |
| PAD24FNCSEL | 1 | 1 | 24 |
| PAD43FNCSEL | 1 | 1 | 43 |
| PAD18FNCSEL | 1 | 2 | 18 |
| PAD25FNCSEL | 1 | 2 | 25 |
| PAD44FNCSEL | 1 | 2 | 44 |
| PAD19FNCSEL | 1 | 3 | 19 |
| PAD26FNCSEL | 1 | 3 | 26 |
| PAD45FNCSEL | 1 | 3 | 45 |
| PAD8FNCSEL | 2 | 4 | 8 |
| PAD16FNCSEL | 1 | 4 | 16 |
| PAD31FNCSEL | 1 | 4 | 31 |
| PAD46FNCSEL | 1 | 4 | 46 |
| PAD4FNCSEL | 2 | 5 | 4 |
| PAD9FNCSEL | 2 | 5 | 9 |
| PAD32FNCSEL | 1 | 5 | 32 |
| PAD47FNCSEL | 1 | 5 | 47 |
| PAD10FNCSEL | 2 | 6 | 10 |
| PAD33FNCSEL | 1 | 6 | 33 |
| PAD48FNCSEL | 1 | 6 | 48 |
| PAD34FNCSEL | 1 | 7 | 34 |
| PAD49FNCSEL | 1 | 7 | 49 |

#### *6.5.1.4 IO Master 1 4-wire SPI Connection*

Four-wire SPI mode of IO Master 1 uses pad 8 as SCK, pad 9 as MISO and pad 10 as MOSI. This mode is configured by setting the PADnFNCSEL fields as shown in Table 226. The PAD8INPEN and PAD9INPEN bits must be set. PAD8PULL, PAD9PULL and PAD10PULL should be cleared.

**Table 226: IO Master 1 4-wire SPI Configuration**

| Field | Value |
|---|---|
| PAD8FNCSEL | 1 |
| PAD9FNCSEL | 1 |
| PAD10FNCSEL | 1 |

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 227. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

Table 227: IO Master 1 4-wire SPI nCE Configuration

| Field | Value | nCE Selection | Pad Used |
|---|---|---|---|
| PAD12FNCSEL | 1 | 0 | 12 |
| PAD35FNCSEL | 1 | 0 | 35 |
| PAD13FNCSEL | 1 | 1 | 13 |
| PAD36FNCSEL | 1 | 1 | 36 |
| PAD14FNCSEL | 1 | 2 | 14 |
| PAD37FNCSEL | 1 | 2 | 37 |
| PAD15FNCSEL | 1 | 3 | 15 |
| PAD38FNCSEL | 1 | 3 | 38 |
| PAD3FNCSEL | 2 | 4 | 3 |
| PAD27FNCSEL | 1 | 4 | 27 |
| PAD20FNCSEL | 2 | 5 | 20 |
| PAD28FNCSEL | 2 | 5 | 28 |
| PAD21FNCSEL | 2 | 6 | 21 |
| PAD29FNCSEL | 1 | 6 | 29 |
| PAD22FNCSEL | 1 | 7 | 22 |
| PAD30FNCSEL | 1 | 7 | 30 |

#### *6.5.1.5 IO Master 0 3-wire SPI Connection*

Three-wire SPI mode of IO Master 0 uses pad 5 as SCK and pad 7 as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 228. The PAD5INPEN and PAD7INPEN bits must be set. PAD5PULL and PAD7PULL should be cleared. Pad 6 may be used for other functions.

Table 228: IO Master 0 3-wire SPI Configuration

| Field | Value |
|---|---|
| PAD5FNCSEL | 1 |
| PAD7FNCSEL | 0 |

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 225. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

#### 6.5.1.6 IO Master 1 3-wire SPI Connection

Three-wire SPI mode of IO Master 1 uses pad 8 as SCK and pad 10 as MOSI/MISO. This mode is configured by setting the PADnFNCSEL fields as shown in Table 229. The PAD8INPEN and PAD10INPEN bits must be set. PAD8PULL and PAD10PULL should be cleared. Pad 9 may be used for other functions.

**Table 229: IO Master 1 3-wire SPI Configuration**

| Field | Value |
|---|---|
| PAD8FNCSEL | 1 |
| PAD10FNCSEL | 0 |

A variety of pads may be used for up to eight nCE signals to select up to eight separate slaves, as shown in Table 227. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

### 6.5.2 Implementing IO Slave Connections

The IO Master module must be correctly connected to the appropriate pads in order to operate.

#### 6.5.2.1 IO Slave I$^2$C Connection

I$^2$C mode of the IO Slave uses pad 0 as SCL and pad 1 as SDA. This mode is configured by setting the PADnFNCSEL fields as shown in Table 230. The PAD0INPEN and PAD1INPEN bits must be set. PAD0PULL and PAD1PULL should be cleared.

**Table 230: IO Slave I$^2$C Configuration**

| Field | Value |
|---|---|
| PAD0FNCSEL | 0 |
| PAD1FNCSEL | 0 |

#### 6.5.2.2 IO Slave 4-wire SPI Connection

Four-wire SPI mode of the IO Slave uses pad 0 as SCK, pad 1 as MISO, pad 2 as MOSI and pad 3 as nCE. This mode is configured by setting the PADnFNCSEL fields as shown in Table 231. The PAD0INPEN, PAD2INPEN and PAD3INPEN bits must be set. PAD0PULL, PAD1PULL, PAD2PULL and PAD3PULL should be cleared.

**Table 231: IO Slave 4-wire SPI Configuration**

| Field | Value |
|---|---|
| PAD0FNCSEL | 1 |
| PAD1FNCSEL | 1 |
| PAD2FNCSEL | 1 |
| PAD3FNCSEL | 1 |

#### 6.5.2.3 IO Slave 3-wire SPI Connection

Three-wire SPI mode of the IO Slave uses pad 0 as SCK, pad 2 as MISO/MOSI and pad 3 as nCE. This mode is configured by setting the PADnFNCSEL fields as shown in Table 232. The PAD0INPEN, PAD2INPEN and PAD3INPEN bits must be set. PAD0PULL, PAD2PULL and PAD3PULL should be cleared. Pad 1 may be used for other functions.

**Table 232: IO Slave 3-wire SPI Configuration**

| Field | Value |
|---|---|
| PAD0FNCSEL | 1 |
| PAD2FNCSEL | 0 |
| PAD3FNCSEL | 1 |

### 6.5.2.4 *IO Slave Interrupt Connection*

The IO Slave can be configured to generate an interrupt output under a variety of internal conditions. If this function is used, the interrupt will be generated on pad 4. PAD4FNCSEL must be set to 1, and PAD4INPEN and PAD4PULL should be cleared.

### 6.5.2.5 *Implementing Internal I$^2$C and SPI Loopback*

The Pad Configuration module includes the capability of implementing internal loopback between the IO Master and IO Slave modules in both I$^2$C and SPI modes. This is useful for testing the Apollo SoC before connecting to external devices. There are six loopback configurations.

#### 6.5.2.5.1 *IO Master 0 I$^2$C Loopback*

In this mode the IO Master 0 module drives the IO Slave in I$^2$C mode. No pullup resistors are enabled, as a logic function implements the wired-OR I$^2$C function, so that all relevant PADnPULL bits should be set to 0. This mode is enabled with the settings in Table 233. PADnINPEN must be set for all of the relevant pads.

**Table 233: IO Master 0 I$^2$C Loopback**

| Field | Value |
|---|---|
| PAD0FNCSEL | 6 |
| PAD1FNCSEL | 6 |
| PAD5FNCSEL | 6 |
| PAD6FNCSEL | 6 |

#### 6.5.2.5.2 *IO Master 1 I$^2$C Loopback*

In this mode the IO Master 1 module drives the IO Slave in I$^2$C mode. No pullup resistors are enabled, as a logic function implements the wired-OR I$^2$C function, so that all relevant PADnPULL bits should be set to 0. This mode is enabled with the settings in Table 234. PADnINPEN must be set for all of the relevant pads.

**Table 234: IO Master 1 I$^2$C Loopback**

| Field | Value |
|---|---|
| PAD0FNCSEL | 7 |
| PAD1FNCSEL | 7 |

Table 234: IO Master 1 $I^2C$ Loopback

| Field | Value |
|---|---|
| PAD8FNCSEL | 7 |
| PAD9FNCSEL | 7 |

#### 6.5.2.5.3 IO Master 0 4-wire SPI Loopback

In this mode the IO Master 0 module drives the IO Slave in 4-wire SPI mode. All relevant PADnPULL bits should be set to 0. This mode is enabled with the settings in Table 235. PADnINPEN must be set for all of the relevant pads.

Table 235: IO Master 0 4-wire SPI Loopback

| Field | Value |
|---|---|
| PAD0FNCSEL | 4 |
| PAD1FNCSEL | 4 |
| PAD2FNCSEL | 4 |
| PAD3FNCSEL | 4 |
| PAD5FNCSEL | 4 |
| PAD6FNCSEL | 4 |

### 6.5.2.6 IO Master 1 4-wire SPI Loopback

In this mode the IO Master 1 module drives the IO Slave in 4-wire SPI mode. All relevant PADnPULL bits should be set to 0. This mode is enabled with the settings in Table 236. PADnINPEN must be set for all of the relevant pads.

Table 236: IO Master 1 4-wire SPI Loopback

| Field | Value |
|---|---|
| PAD0FNCSEL | 5 |
| PAD1FNCSEL | 5 |
| PAD2FNCSEL | 5 |
| PAD3FNCSEL | 5 |
| PAD8FNCSEL | 5 |
| PAD9FNCSEL | 5 |

### 6.5.2.7 IO Master 0 3-wire SPI Loopback

In this mode the IO Master 0 module drives the IO Slave in 3-wire SPI mode. All relevant PADnPULL bits should be set to 0. This mode is enabled with the settings in Table 237. PADnINPEN must be set for all of the relevant pads.

Table 237: IO Master 0 3-wire SPI Loopback

| Field | Value |
|---|---|
| PAD0FNCSEL | 4 |

Table 237: IO Master 0 3-wire SPI Loopback

| Field | Value |
|---|---|
| PAD2FNCSEL | 6 |
| PAD3FNCSEL | 4 |
| PAD5FNCSEL | 4 |
| PAD7FNCSEL | 6 |

#### *6.5.2.8 IO Master 1 3-wire SPI Loopback*

In this mode the IO Master 1 module drives the IO Slave in 3-wire SPI mode. All relevant PADnPULL bits should be set to 0. This mode is enabled with the settings in Table 238. PADnINPEN must be set for all of the relevant pads.

Table 238: IO Master 1 3-wire SPI Loopback

| Field | Value |
|---|---|
| PAD0FNCSEL | 5 |
| PAD2FNCSEL | 7 |
| PAD3FNCSEL | 5 |
| PAD8FNCSEL | 5 |
| PAD10FNCSEL | 7 |

#### *6.5.2.9 IO Slave Interrupt Loopback*

The interrupt output from the IO Slave may be looped back to a GPIO (GPIO4). This is accomplished by setting PAD4FNCSEL to 4, and monitoring the interrupt output by reading GPIO4. PAD4INPEN must be set.

### *6.5.3 Implementing Counter/Timer Connections*

Each Counter/Timer can optionally count pulses from an input pad, or generate pulses on an output pad. Table 239 shows the PADnFNCSEL settings to connect each Counter/Timer to the appropriate pad. If the pad is used as an input, the PADnINPEN bit should be set, otherwise it should be cleared. The PADnPULL bit may be set if the input signal is open drain.

Table 239: Counter/Timer Pad Configuration

| Field | Value | Ctr/Timer | Pad |
|---|---|---|---|
| PAD12FNCSEL | 2 | A0 | 12 |
| PAD25FNCSEL | 2 | A0 | 25 |
| PAD42FNCSEL | 2 | A0 | 42 |
| PAD13FNCSEL | 2 | B0 | 13 |
| PAD26FNCSEL | 2 | B0 | 26 |
| PAD43FNCSEL | 2 | B0 | 43 |
| PAD18FNCSEL | 2 | A1 | 18 |
| PAD27FNCSEL | 2 | A1 | 27 |
| PAD44FNCSEL | 2 | A1 | 44 |
| PAD19FNCSEL | 2 | B1 | 19 |
| PAD28FNCSEL | 2 | B1 | 28 |
| PAD45FNCSEL | 2 | B1 | 45 |
| PAD20FNCSEL | 2 | A2 | 20 |
| PAD29FNCSEL | 2 | A2 | 29 |
| PAD46FNCSEL | 2 | A2 | 46 |
| PAD21FNCSEL | 2 | B2 | 21 |
| PAD30FNCSEL | 2 | B2 | 30 |
| PAD47FNCSEL | 2 | B2 | 47 |
| PAD22FNCSEL | 2 | A3 | 22 |
| PAD31FNCSEL | 2 | A3 | 31 |
| PAD48FNCSEL | 2 | A3 | 48 |
| PAD23FNCSEL | 2 | B3 | 23 |
| PAD32FNCSEL | 2 | B3 | 32 |
| PAD49FNCSEL | 2 | B3 | 49 |

### *6.5.4 Implementing UART Connections*

The UART signals can be connected to a variety of pads.

#### *6.5.4.1 UART TX/RX Connections*

The UART data signals TX and RX may each be connected to several pads. Note that TX and RX are selected independently. Table 240 shows the connections for TX, which should have the corresponding PADnINPEN and PADnPULL fields clear. Table 241 shows the connections for RX, which must have the corresponding PADnINPEN field set and should have the corresponding PADnPULL field clear.

**Table 240: UART TX Configuration**

| Field | Value | Pad |
|---|---|---|
| PAD0FNCSEL | 2 | 0 |
| PAD14FNCSEL | 2 | 14 |
| PAD22FNCSEL | 0 | 22 |
| PAD35FNCSEL | 2 | 35 |
| PAD39FNCSEL | 1 | 39 |

**Table 241: UART RX Configuration**

| Field | Value | Pad |
|---|---|---|
| PAD1FNCSEL | 2 | 1 |
| PAD15FNCSEL | 2 | 15 |
| PAD23FNCSEL | 0 | 23 |
| PAD36FNCSEL | 2 | 36 |
| PAD40FNCSEL | 1 | 40 |

#### 6.5.4.2 UART RTS/CTS Connections

The UART modem control signals RTS and CTS may each be connected to one of two pads. Note that RTS and CTS are selected independently. Table 242 shows the connections for RTS, which should have the corresponding PADnINPEN and PADnPULL fields clear. Table 243 shows the connections for CTS, which must have the corresponding PADnINPEN field set and should have the corresponding PADnPULL field clear.

**Table 242: UART RTS Configuration**

| Field | Value | Pad |
|---|---|---|
| PAD5FNCSEL | 2 | 5 |
| PAD37FNCSEL | 2 | 37 |

**Table 243: UART CTS Configuration**

| Field | Value | Pad |
|---|---|---|
| PAD6FNCSEL | 2 | 6 |
| PAD38FNCSEL | 2 | 38 |

### 6.5.5 Implementing GPIO Connections

Each pad of the Apollo SoC can be configured as a GPIO port by setting PADnFNCSEL to 3. PADnINPEN and PADnPULL must be set appropriately depending on the specific GPIO function.

### 6.5.6 Implementing CLKOUT Connections

The flexible clock output of the Clock Generator module, CLKOUT, may be configured on several pads as shown in Table 244. PADnINPEN and PADnPULL should be cleared in each case.

**Table 244: CLKOUT Configuration**

| Field | Value | Pad |
|---|---|---|
| PAD2FNCSEL | 2 | 2 |
| PAD4FNCSEL | 6 | 4 |
| PAD7FNCSEL | 2 | 7 |
| PAD11FNCSEL | 2 | 11 |
| PAD24FNCSEL | 2 | 24 |
| PAD39FNCSEL | 2 | 39 |

### 6.5.7 Implementing ADC Connections

Three types of pad connections may be made for the ADC module. Up to eight pads may be selected from and configured as the analog inputs, as shown in Table 245. The ADCREF reference voltage input supplied on pad 16 is configured by setting PAD16FNCSEL to 0.

If an external digital trigger is desired, up to twelve selectable pad choices may be selected from and configured, as shown in Table 246. For the trigger inputs, PADnINPEN must be set. For other inputs, PADnINPEN should be cleared. PADnPULL should be cleared except in the case of an open drain trigger input.

**Table 245: ADC Analog Input Configuration**

| Field | Value | Input | Pad |
|---|---|---|---|
| PAD12FNCSEL | 0 | ADC0 | 12 |
| PAD13FNCSEL | 0 | ADC1 | 13 |
| PAD14FNCSEL | 0 | ADC2 | 14 |
| PAD15FNCSEL | 0 | ADC3 | 15 |
| PAD29FNCSEL | 0 | ADC4 | 29 |
| PAD30FNCSEL | 0 | ADC5 | 30 |
| PAD31FNCSEL | 0 | ADC6 | 31 |
| PAD32FNCSEL | 0 | ADC7 | 32 |

**Table 246: ADC Trigger Input Configuration**

| Field | Value | Input | Pad |
|---|---|---|---|
| PAD3FNCSEL | 0 | TRIG0 | 3 |
| PAD37FNCSEL | 0 | TRIG0 | 37 |
| PAD4FNCSEL | 0 | TRIG1 | 4 |
| PAD38FNCSEL | 0 | TRIG1 | 38 |
| PAD16FNCSEL | 0 | TRIG2 | 16 |
| PAD39FNCSEL | 0 | TRIG2 | 39 |
| PAD17FNCSEL | 0 | TRIG3 | 17 |
| PAD40FNCSEL | 0 | TRIG3 | 40 |
| PAD41FNCSEL | 0 | TRIG4 | 41 |
| PAD42FNCSEL | 0 | TRIG5 | 42 |
| PAD43FNCSEL | 0 | TRIG6 | 43 |
| PAD44FNCSEL | 0 | TRIG7 | 44 |

### *6.5.8 Implementing Voltage Comparator Connections*

Two types of pad connections may be made for the Voltage Comparator (VCOMP) module. Three reference voltages may be used for the comparator negative input as shown in Table 247. The voltage to be applied to the comparator positive input are shown in Table 248. In each case PADnINPENn and PADnPULL should be cleared.

**Table 247: Voltage Comparator Reference Configuration**

| Field | Value | Input | Pad |
|---|---|---|---|
| PAD19FNCSEL | 0 | CMPRF0 | 19 |
| PAD33FNCSEL | 0 | CMPRF1 | 33 |
| PAD34FNCSEL | 0 | CMPRF2 | 34 |

**Table 248: Voltage Comparator Input Configuration**

| Field | Value | Input | Pad |
|---|---|---|---|
| PAD17FNCSEL | 0 | CMPIN0 | 17 |
| PAD18FNCSEL | 0 | CMPIN1 | 18 |

### *6.5.9 Implementing the Software Debug Port Connections*

The software debug clock (SWDCK) and data (SWDIO) must be connected on pads 20 and 21 respectively. PAD20FNCSEL and PAD21FNCSEL must be set to 0, PAD20INPEN and PAD21INPEN must be set, and PAD20PULL and PAD21PULL must be set, which results in a default state of SWDCK low and SWDIO high.

The optional continuous output signal SWO may be configured on a variety of pads as shown in Table 249, and PADnINPEN and PADnPULL should be cleared for the selected pad.

**Table 249: SWO Configuration**

| Field | Value | Pad |
|---|---|---|
| PAD4FNCSEL | 5 | 4 |
| PAD13FNCSEL | 6 | 13 |
| PAD41FNCSEL | 2 | 41 |

## 6.6 GPIO Registers

**General Purpose IO**

**INSTANCE 0 BASE ADDRESS:**0x40010000

This is the detailed description of the general purpose I/O (GPIO) block, as well as for the PAD multiplexor. Note that GPIO interrupt bits are edge triggered. WARNING: if an interrupt bit is cleared while the combination of polarity and input are still asserted then this bit will not set again.

### *6.6.1 Register Memory Map*

**Table 250: GPIO Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0x40010000 | PADREGA | Pad Configuration Register A |
| 0x40010004 | PADREGB | Pad Configuration Register B |
| 0x40010008 | PADREGC | Pad Configuration Register C |
| 0x4001000C | PADREGD | Pad Configuration Register D |
| 0x40010010 | PADREGE | Pad Configuration Register E |
| 0x40010014 | PADREGF | Pad Configuration Register F |
| 0x40010018 | PADREGG | Pad Configuration Register G |
| 0x4001001C | PADREGH | Pad Configuration Register H |
| 0x40010020 | PADREGI | Pad Configuration Register I |
| 0x40010024 | PADREGJ | Pad Configuration Register J |
| 0x40010028 | PADREGK | Pad Configuration Register K |
| 0x4001002C | PADREGL | Pad Configuration Register L |
| 0x40010030 | PADREGM | Pad Configuration Register M |
| 0x40010040 | CFGA | GPIO Configuration Register A |
| 0x40010044 | CFGB | GPIO Configuration Register B |
| 0x40010048 | CFGC | GPIO Configuration Register C |
| 0x4001004C | CFGD | GPIO Configuration Register D |
| 0x40010050 | CFGE | GPIO Configuration Register E |
| 0x40010054 | CFGF | GPIO Configuration Register F |
| 0x40010058 | CFGG | GPIO Configuration Register G |
| 0x40010060 | PADKEY | Key Register for all pad configuration registers |
| 0x40010080 | RDA | GPIO Input Register A |
| 0x40010084 | RDB | GPIO Input Register B |
| 0x40010088 | WTA | GPIO Output Register A |
| 0x4001008C | WTB | GPIO Output Register B |
| 0x40010090 | WTSA | GPIO Output Register A Set |
| 0x40010094 | WTSB | GPIO Output Register B Set |
| 0x40010098 | WTCA | GPIO Output Register A Clear |
| 0x4001009C | WTCB | GPIO Output Register B Clear |
| 0x400100A0 | ENA | GPIO Enable Register A |

**Table 250: GPIO Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0x400100A4 | ENB | GPIO Enable Register B |
| 0x400100A8 | ENSA | GPIO Enable Register A Set |
| 0x400100AC | ENSB | GPIO Enable Register B Set |
| 0x400100B4 | ENCA | GPIO Enable Register A Clear |
| 0x400100B8 | ENCB | GPIO Enable Register B Clear |
| 0x40010200 | INT0EN | GPIO Interrupt Registers 31-0: Enable |
| 0x40010204 | INT0STAT | GPIO Interrupt Registers 31-0: Status |
| 0x40010208 | INT0CLR | GPIO Interrupt Registers 31-0: Clear |
| 0x4001020C | INT0SET | GPIO Interrupt Registers 31-0: Set |
| 0x40010210 | INT1EN | GPIO Interrupt Registers 49-32: Enable |
| 0x40010214 | INT1STAT | GPIO Interrupt Registers 49-32: Status |
| 0x40010218 | INT1CLR | GPIO Interrupt Registers 49-32: Clear |
| 0x4001021C | INT1SET | GPIO Interrupt Registers 49-32: Set |

## 6.6.2 *GPIO Registers*

### 6.6.2.1 *PADREGA Register*

**Pad Configuration Register A**

**OFFSET:** 0x00000000

**INSTANCE 0 ADDRESS:** 0x40010000

This register controls the pad configuration controls for PAD3 through PAD0. Writes to this register must be unlocked by the PADKEY register.

**Table 251: PADREGA Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PAD3PWRUP | RSVD | PAD3FNCSEL | | | PAD3STRNG | PAD3INPEN | PAD3PULL | RSVD | | PAD2FNCSEL | | | PAD2STRNG | PAD2INPEN | PAD2PULL | RSVD | | PAD1FNCSEL | | | PAD1STRNG | PAD1INPEN | PAD1PULL | RSVD | | PAD0FNCSEL | | | PAD0STRNG | PAD0INPEN | PAD0PULL |

### Table 252: PADREGA Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | PAD3PWRUP | 0x0 | RW | Pad 3 upper power switch enable<br><br>DIS = 0x0 - Power switch disabled<br>EN = 0x1 - Power switch enabled |
| 30 | RSVD | 0x0 | RO | RESERVED |
| 29:27 | PAD3FNCSEL | 0x3 | RW | Pad 3 function select<br><br>TRIG0 = 0x0 - Configure as the ADC Trigger 0 signal<br>SLnCE = 0x1 - Configure as the IOSLAVE SPI nCE signal<br>M1nCE4 = 0x2 - Configure as the SPI channel 4 nCE signal from IOMSTR1<br>GPIO3 = 0x3 - Configure as GPIO3<br>M0nCE = 0x4 - Configure as the IOSLAVE SPI nCE loopback signal from IOMSTR0<br>M1nCE = 0x5 - Configure as the IOSLAVE SPI nCE loopback signal from IOMSTR1<br>DIS = 0x6 - Pad disabled |
| 26 | PAD3STRNG | 0x0 | RW | Pad 3 drive strength.<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 25 | PAD3INPEN | 0x0 | RW | Pad 3 input enable.<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 24 | PAD3PULL | 0x0 | RW | Pad 3 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 23:22 | RSVD | 0x0 | RO | RESERVED |
| 21:19 | PAD2FNCSEL | 0x3 | RW | Pad 2 function select<br><br>SLWIR3 = 0x0 - Configure as the IOSLAVE SPI 3-wire MOSI/MISO signal<br>SLMOSI = 0x1 - Configure as the IOSLAVE SPI MOSI signal<br>CLKOUT = 0x2 - Configure as the CLKOUT signal<br>GPIO2 = 0x3 - Configure as GPIO2<br>M0MOSI = 0x4 - Configure as the IOSLAVE SPI MOSI loopback signal from IOMSTR0<br>M1MOSI = 0x5 - Configure as the IOSLAVE SPI MOSI loopback signal from IOMSTR1<br>M0WIR3 = 0x6 - Configure as the IOSLAVE SPI 3-wire MOSI/MISO loopback signal from IOMSTR0<br>M1WIR3 = 0x7 - Configure as the IOSLAVE SPI 3-wire MOSI/MISO loopback signal from IOMSTR1 |
| 18 | PAD2STRNG | 0x0 | RW | Pad 2 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 17 | PAD2INPEN | 0x0 | RW | Pad 2 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |

### Table 252: PADREGA Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 16 | PAD2PULL | 0x0 | RW | Pad 2 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 15:14 | RSVD | 0x0 | RO | RESERVED |
| 13:11 | PAD1FNCSEL | 0x3 | RW | Pad 1 function select<br><br>SLSDA = 0x0 - Configure as the IOSLAVE I2C SDA signal<br>SLMISO = 0x1 - Configure as the IOSLAVE SPI MISO signal<br>UARTRX = 0x2 - Configure as the UART RX signal<br>GPIO1 = 0x3 - Configure as GPIO1<br>M0MISO = 0x4 - Configure as the IOSLAVE SPI MISO loopback signal from IOMSTR0<br>M1MISO = 0x5 - Configure as the IOSLAVE SPI MISO loopback signal from IOMSTR1<br>M0SDA = 0x6 - Configure as the IOSLAVE I2C SDA loopback signal from IOMSTR0<br>M1SDA = 0x7 - Configure as the IOSLAVE I2C SDA loopback signal from IOMSTR1 |
| 10 | PAD1STRNG | 0x0 | RW | Pad 1 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 9 | PAD1INPEN | 0x0 | RW | Pad 1 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD1PULL | 0x0 | RW | Pad 1 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7:6 | RSVD | 0x0 | RO | RESERVED |
| 5:3 | PAD0FNCSEL | 0x3 | RW | Pad 0 function select<br><br>SLSCL = 0x0 - Configure as the IOSLAVE I2C SCL signal<br>SLSCK = 0x1 - Configure as the IOSLAVE SPI SCK signal<br>UARTTX = 0x2 - Configure as the UART TX signal<br>GPIO0 = 0x3 - Configure as GPIO0<br>M0SCK = 0x4 - Configure as the IOSLAVE SPI SCK loopback signal from IOMSTR0<br>M1SCK = 0x5 - Configure as the IOSLAVE SPI SCK loopback signal from IOMSTR1<br>M0SCL = 0x6 - Configure as the IOSLAVE I2C SCL loopback signal from IOMSTR0<br>M1SCL = 0x7 - Configure as the IOSLAVE I2C SCL loopback signal from IOMSTR1 |
| 2 | PAD0STRNG | 0x0 | RW | Pad 0 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |

**Table 252: PADREGA Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 1 | PAD0INPEN | 0x0 | RW | Pad 0 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD0PULL | 0x0 | RW | Pad 0 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |

### 6.6.2.2 *PADREGB Register*

**Pad Configuration Register B**

**OFFSET:** 0x00000004

**INSTANCE 0 ADDRESS:** 0x40010004

This register controls the pad configuration controls for PAD7 through PAD4. Writes to this register must be unlocked by the PADKEY register.

**Table 253: PADREGB Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | PAD7FNCSEL | | | PAD7STRNG | PAD7INPEN | PAD7PULL | PAD6RSEL | | PAD6FNCSEL | | | PAD6STRNG | PAD6INPEN | PAD6PULL | PAD5RSEL | | PAD5FNCSEL | | | PAD5STRNG | PAD5INPEN | PAD5PULL | PAD4PWRUP | RSVD | PAD4FNCSEL | | | PAD4STRNG | PAD4INPEN | PAD4PULL |

**Table 254: PADREGB Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:30 | RSVD | 0x0 | RO | RESERVED |
| 29:27 | PAD7FNCSEL | 0x3 | RW | Pad 7 function select<br><br>M0WIR3 = 0x0 - Configure as the IOMSTR0 SPI 3-wire MOSI/MISO signal<br>M0MOSI = 0x1 - Configure as the IOMSTR0 SPI MOSI signal<br>CLKOUT = 0x2 - Configure as the CLKOUT signal<br>GPIO7 = 0x3 - Configure as GPIO7<br>SLWIR3 = 0x6 - Configure as the IOMSTR0 SPI 3-wire MOSI/MISO loop-back signal from IOSLAVE<br>DIS = 0x7 - Pad disabled |
| 26 | PAD7STRNG | 0x0 | RW | Pad 7 drive strentgh<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |

### Table 254: PADREGB Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 25 | PAD7INPEN | 0x0 | RW | Pad 7 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 24 | PAD7PULL | 0x0 | RW | Pad 7 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 23:22 | PAD6RSEL | 0x0 | RW | Pad 6 pullup resistor selection.<br><br>PULL1_5K = 0x0 - Pullup is ~1.5 KOhms<br>PULL6K = 0x1 - Pullup is ~6 KOhms<br>PULL12K = 0x2 - Pullup is ~12 KOhms<br>PULL24K = 0x3 - Pullup is ~24 KOhms |
| 21:19 | PAD6FNCSEL | 0x3 | RW | Pad 6 function select<br><br>M0SDA = 0x0 - Configure as the IOMSTR0 I2C SDA signal<br>M0MISO = 0x1 - Configure as the IOMSTR0 SPI MISO signal<br>UACTS = 0x2 - Configure as the UART CTS signal<br>GPIO6 = 0x3 - Configure as GPIO6<br>SLMISO = 0x4 - Configure as the IOMSTR0 SPI MISO loopback signal from IOSLAVE<br>SLSDA = 0x6 - Configure as the IOMSTR0 I2C SDA loopback signal from IOSLAVE<br>DIS = 0x7 - Pad disabled |
| 18 | PAD6STRNG | 0x0 | RW | Pad 6 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 17 | PAD6INPEN | 0x0 | RW | Pad 6 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 16 | PAD6PULL | 0x0 | RW | Pad 6 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 15:14 | PAD5RSEL | 0x0 | RW | Pad 5 pullup resistor selection.<br><br>PULL1_5K = 0x0 - Pullup is ~1.5 KOhms<br>PULL6K = 0x1 - Pullup is ~6 KOhms<br>PULL12K = 0x2 - Pullup is ~12 KOhms<br>PULL24K = 0x3 - Pullup is ~24 KOhms |
| 13:11 | PAD5FNCSEL | 0x3 | RW | Pad 5 function select<br><br>M0SCL = 0x0 - Configure as the IOMSTR0 I2C SCL signal<br>M0SCK = 0x1 - Configure as the IOMSTR0 SPI SCK signal<br>UARTS = 0x2 - Configure as the UART RTS signal<br>GPIO5 = 0x3 - Configure as GPIO5<br>SLSCK = 0x4 - Configure as the IOMSTR0 SPI SCK loopback signal from IOSLAVE<br>SLSCL = 0x6 - Configure as the IOMSTR0 I2C SCL loopback signal from IOSLAVE<br>DIS = 0x7 - Pad disabled |

Table 254: PADREGB Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 10 | PAD5STRNG | 0x0 | RW | Pad 5 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 9 | PAD5INPEN | 0x0 | RW | Pad 5 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD5PULL | 0x0 | RW | Pad 5 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7 | PAD4PWRUP | 0x0 | RW | Pad 4 upper power switch enable<br><br>DIS = 0x0 - Power switch disabled<br>EN = 0x1 - Power switch enabled |
| 6 | RSVD | 0x0 | RO | RESERVED |
| 5:3 | PAD4FNCSEL | 0x3 | RW | Pad 4 function select<br><br>TRIG1 = 0x0 - Configure as the ADC Trigger 1 signal<br>SLINT = 0x1 - Configure as the IOSLAVE interrupt out signal<br>M0nCE5 = 0x2 - Configure as the SPI channel 5 nCE signal from IOMSTR0<br>GPIO4 = 0x3 - Configure as GPIO4<br>SLINTGP = 0x4 - Configure as the IOSLAVE interrupt loopback signal to GPIO4<br>SWO = 0x5 - Configure as the serial wire debug SWO signal<br>CLKOUT = 0x6 - Configure as the CLKOUT signal<br>DIS = 0x7 - Pad disabled |
| 2 | PAD4STRNG | 0x0 | RW | Pad 4 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 1 | PAD4INPEN | 0x0 | RW | Pad 4 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD4PULL | 0x0 | RW | Pad 4 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |

### *6.6.2.3 PADREGC Register*

**Pad Configuration Register C**

**OFFSET:** 0x00000008

**INSTANCE 0 ADDRESS:** 0x40010008

This register controls the pad configuration controls for PAD11 through PAD8. Writes to this register must be unlocked by the PADKEY register.

Table 255: PADREGC Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | PAD11PWRDN | RSVD | PAD11FNCSEL | | PAD11STRNG | PAD11INPEN | PAD11PULL | RSVD | | PAD10FNCSEL | | | PAD10STRNG | PAD10INPEN | PAD10PULL | PAD9RSEL | | PAD9FNCSEL | | | PAD9STRNG | PAD9INPEN | PAD9PULL | PAD8RSEL | | PAD8FNCSEL | | | PAD8STRNG | PAD8INPEN | PAD8PULL |

Table 256: PADREGC Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | RSVD | 0x0 | RO | RESERVED |
| 30 | PAD11PWRDN | 0x0 | RW | Pad 11 lower power switch enable<br><br>DIS = 0x0 - Power switch disabled<br>EN = 0x1 - Power switch enabled |
| 29 | RSVD | 0x0 | RO | RESERVED |
| 28:27 | PAD11FNCSEL | 0x3 | RW | Pad 11 function select<br><br>ANATST = 0x0 - Configure as the analog test output signal<br>M0nCE0 = 0x1 - Configure as the SPI channel 0 nCE signal from IOMSTR0<br>CLKOUT = 0x2 - Configure as the CLKOUT signal<br>GPIO11 = 0x3 - Configure as GPIO11 |
| 26 | PAD11STRNG | 0x0 | RW | Pad 11 drive strentgh<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 25 | PAD11INPEN | 0x0 | RW | Pad 11 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 24 | PAD11PULL | 0x0 | RW | Pad 11 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 23:22 | RSVD | 0x0 | RO | RESERVED |
| 21:19 | PAD10FNCSEL | 0x3 | RW | Pad 10 function select<br><br>M1WIR3 = 0x0 - Configure as the IOMSTR1 SPI 3-wire MOSI/MISO signal<br>M1MOSI = 0x1 - Configure as the IOMSTR1 SPI MOSI signal<br>M0nCE6 = 0x2 - Configure as the SPI channel 6 nCE signal from IOMSTR0<br>GPIO10 = 0x3 - Configure as GPIO10<br>DIS = 0x6 - Pad disabled<br>SLWIR3 = 0x7 - Configure as the IOMSTR1 SPI 3-wire MOSI/MISO loopback signal from IOSLAVE |

**Table 256: PADREGC Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 18 | PAD10STRNG | 0x0 | RW | Pad 10 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 17 | PAD10INPEN | 0x0 | RW | Pad 10 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 16 | PAD10PULL | 0x0 | RW | Pad 10 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 15:14 | PAD9RSEL | 0x0 | RW | Pad 9 pullup resistor selection<br><br>PULL1_5K = 0x0 - Pullup is ~1.5 KOhms<br>PULL6K = 0x1 - Pullup is ~6 KOhms<br>PULL12K = 0x2 - Pullup is ~12 KOhms<br>PULL24K = 0x3 - Pullup is ~24 KOhms |
| 13:11 | PAD9FNCSEL | 0x3 | RW | Pad 9 function select<br><br>M1SDA = 0x0 - Configure as the IOMSTR1 I2C SDA signal<br>M1MISO = 0x1 - Configure as the IOMSTR1 SPI MISO signal<br>M0nCE5 = 0x2 - Configure as the SPI channel 5 nCE signal from IOMSTR0<br>GPIO9 = 0x3 - Configure as GPIO9<br>SLMISO = 0x5 - Configure as the IOMSTR1 SPI MISO loopback signal from IOSLAVE<br>DIS = 0x6 - Pad disabled<br>SLSDA = 0x7 - Configure as the IOMSTR1 I2C SDA loopback signal from IOSLAVE |
| 10 | PAD9STRNG | 0x0 | RW | Pad 9 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 9 | PAD9INPEN | 0x0 | RW | Pad 9 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD9PULL | 0x0 | RW | Pad 9 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7:6 | PAD8RSEL | 0x0 | RW | Pad 8 pullup resistor selection.<br><br>PULL1_5K = 0x0 - Pullup is ~1.5 KOhms<br>PULL6K = 0x1 - Pullup is ~6 KOhms<br>PULL12K = 0x2 - Pullup is ~12 KOhms<br>PULL24K = 0x3 - Pullup is ~24 KOhms |

**Table 256: PADREGC Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 5:3 | PAD8FNCSEL | 0x3 | RW | Pad 8 function select<br><br>M1SCL = 0x0 - Configure as the IOMSTR1 I2C SCL signal<br>M1SCK = 0x1 - Configure as the IOMSTR1 SPI SCK signal<br>M0nCE4 = 0x2 - Configure as the SPI channel 4 nCE signal from IOMSTR0<br>GPIO8 = 0x3 - Configure as GPIO8<br>SLSCK = 0x5 - Configure as the IOMSTR1 SPI SCK loopback signal from IOSLAVE<br>DIS = 0x6 - Pad disabled<br>SLSCL = 0x7 - Configure as the IOMSTR1 I2C SCL loopback signal from IOSLAVE |
| 2 | PAD8STRNG | 0x0 | RW | Pad 8 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 1 | PAD8INPEN | 0x0 | RW | Pad 8 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD8PULL | 0x0 | RW | Pad 8 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |

### 6.6.2.4 *PADREGD Register*

**Pad Configuration Register D**

**OFFSET:** 0x0000000C

**INSTANCE 0 ADDRESS:** 0x4001000C

This register controls the pad configuration controls for PAD15 through PAD12. Writes to this register must be unlocked by the PADKEY register.

**Table 257: PADREGD Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | PAD15FNCSEL | | | PAD15STRNG | PAD15INPEN | PAD15PULL | RSVD | | PAD14FNCSEL | | | PAD14STRNG | PAD14INPEN | PAD14PULL | RSVD | | PAD13FNCSEL | | | PAD13STRNG | PAD13INPEN | PAD13PULL | RSVD | | | PAD12FNCSEL | | PAD12STRNG | PAD12INPEN | PAD12PULL |

**Table 258: PADREGD Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:30 | RSVD | 0x0 | RO | RESERVED |

Table 258: PADREGD Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 29:27 | PAD15FNCSEL | 0x3 | RW | Pad 15 function select<br><br>ADC3 = 0x0 - Configure as the analog ADC input 3<br>M1nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR1<br>UARTRX = 0x2 - Configure as the UART RX signal<br>GPIO15 = 0x3 - Configure as GPIO15<br>DIS = 0x7 - Pad disabled |
| 26 | PAD15STRNG | 0x0 | RW | Pad 15 drive strentgh<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 25 | PAD15INPEN | 0x0 | RW | Pad 15 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 24 | PAD15PULL | 0x0 | RW | Pad 15 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 23:22 | RSVD | 0x0 | RO | RESERVED |
| 21:19 | PAD14FNCSEL | 0x3 | RW | Pad 14 function select<br><br>ADC2 = 0x0 - Configure as the analog ADC input 2<br>M1nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR1<br>UARTTX = 0x2 - Configure as the UART TX signal<br>GPIO14 = 0x3 - Configure as GPIO14<br>DIS = 0x7 - Pad disabled |
| 18 | PAD14STRNG | 0x0 | RW | Pad 14 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 17 | PAD14INPEN | 0x0 | RW | Pad 14 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 16 | PAD14PULL | 0x0 | RW | Pad 14 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 15:14 | RSVD | 0x0 | RO | RESERVED |
| 13:11 | PAD13FNCSEL | 0x3 | RW | Pad 13 function select<br><br>ADC1 = 0x0 - Configure as the analog ADC input 1<br>M1nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR1<br>TCTB0 = 0x2 - Configure as the input/output signal from CTIMER B0<br>GPIO13 = 0x3 - Configure as GPIO13<br>SWO = 0x6 - Configure as the serial wire debug SWO signal<br>DIS = 0x7 - Pad disabled |
| 10 | PAD13STRNG | 0x0 | RW | Pad 13 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |

**Table 258: PADREGD Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 9 | PAD13INPEN | 0x0 | RW | Pad 13 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD13PULL | 0x0 | RW | Pad 13 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7:5 | RSVD | 0x0 | RO | RESERVED |
| 4:3 | PAD12FNCSEL | 0x3 | RW | Pad 12 function select<br><br>ADC0 = 0x0 - Configure as the analog ADC input 0<br>M1nCE0 = 0x1 - Configure as the SPI channel 0 nCE signal from IOMSTR1<br>TCTA0 = 0x2 - Configure as the input/output signal from CTIMER A0<br>GPIO12 = 0x3 - Configure as GPIO12 |
| 2 | PAD12STRNG | 0x0 | RW | Pad 12 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 1 | PAD12INPEN | 0x0 | RW | Pad 12 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD12PULL | 0x0 | RW | Pad 12 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |

### *6.6.2.5 PADREGE Register*

**Pad Configuration Register E**

**OFFSET:** 0x00000010

**INSTANCE 0 ADDRESS:** 0x40010010

This register controls the pad configuration controls for PAD19 through PAD16. Writes to this register must be unlocked by the PADKEY register.

**Table 259: PADREGE Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | PAD19FNCSEL | | PAD19STRNG | PAD19INPEN | PAD19PULL | RSVD | | | PAD18FNCSEL | | PAD18STRNG | PAD18INPEN | PAD18PULL | RSVD | | PAD17FNCSEL | | | PAD17STRNG | PAD17INPEN | PAD17PULL | RSVD | | | PAD16FNCSEL | | PAD16STRNG | PAD16INPEN | PAD16PULL |

### Table 260: PADREGE Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:29 | RSVD | 0x0 | RO | RESERVED |
| 28:27 | PAD19FNCSEL | 0x3 | RW | Pad 19 function select<br><br>CMPRF = 0x0 - Configure as the analog comparator reference signal<br>M0nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR0<br>TCTB1 = 0x2 - Configure as the input/output signal from CTIMER B1<br>GPIO19 = 0x3 - Configure as GPIO19 |
| 26 | PAD19STRNG | 0x0 | RW | Pad 19 drive strentgh<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 25 | PAD19INPEN | 0x0 | RW | Pad 19 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 24 | PAD19PULL | 0x0 | RW | Pad 19 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 23:21 | RSVD | 0x0 | RO | RESERVED |
| 20:19 | PAD18FNCSEL | 0x3 | RW | Pad 18 function select<br><br>CMPIN1 = 0x0 - Configure as the analog comparator input 1 signal<br>M0nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR0<br>TCTA1 = 0x2 - Configure as the input/output signal from CTIMER A1<br>GPIO18 = 0x3 - Configure as GPIO18 |
| 18 | PAD18STRNG | 0x0 | RW | Pad 18 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 17 | PAD18INPEN | 0x0 | RW | Pad 18 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 16 | PAD18PULL | 0x0 | RW | Pad 18 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 15:14 | RSVD | 0x0 | RO | RESERVED |
| 13:11 | PAD17FNCSEL | 0x3 | RW | Pad 17 function select<br><br>CMPIN0 = 0x0 - Configure as the analog comparator input 0 signal<br>M0nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR0<br>TRIG3 = 0x2 - Configure as the ADC Trigger 3 signal<br>GPIO17 = 0x3 - Configure as GPIO17<br>DIS = 0x7 - Pad disabled |

**Table 260: PADREGE Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 10 | PAD17STRNG | 0x0 | RW | Pad 17 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 9 | PAD17INPEN | 0x0 | RW | Pad 17 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD17PULL | 0x0 | RW | Pad 17 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7:5 | RSVD | 0x0 | RO | RESERVED |
| 4:3 | PAD16FNCSEL | 0x3 | RW | Pad 16 function select<br><br>ADCREF = 0x0 - Configure as the analog ADC reference input signal<br>M0nCE4 = 0x1 - Configure as the SPI channel 4 nCE signal from IOMSTR0<br>TRIG2 = 0x2 - Configure as the ADC Trigger 2 signal<br>GPIO16 = 0x3 - Configure as GPIO16 |
| 2 | PAD16STRNG | 0x0 | RW | Pad 16 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 1 | PAD16INPEN | 0x0 | RW | Pad 16 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD16PULL | 0x0 | RW | Pad 16 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |

### *6.6.2.6 PADREGF Register*

**Pad Configuration Register F**

**OFFSET:** 0x00000014

**INSTANCE 0 ADDRESS:** 0x40010014

This register controls the pad configuration controls for PAD23 through PAD20. Writes to this register must be unlocked by the PADKEY register.

**Table 261: PADREGF Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | PAD23FNCSEL | | PAD23STRNG | PAD23INPEN | PAD23PULL | RSVD | | | PAD22FNCSEL | | PAD22STRNG | PAD22INPEN | PAD22PULL | RSVD | | | PAD21FNCSEL | | PAD21STRNG | PAD21INPEN | PAD21PULL | RSVD | | | PAD20FNCSEL | | PAD20STRNG | PAD20INPEN | PAD20PULL |

**Table 262: PADREGF Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:29 | RSVD | 0x0 | RO | RESERVED |
| 28:27 | PAD23FNCSEL | 0x3 | RW | Pad 23 function select<br><br>UARTRX = 0x0 - Configure as the UART RX signal<br>M0nCE0 = 0x1 - Configure as the SPI channel 0 nCE signal from IOMSTR0<br>TCTB3 = 0x2 - Configure as the input/output signal from CTIMER B3<br>GPIO23 = 0x3 - Configure as GPIO23 |
| 26 | PAD23STRNG | 0x0 | RW | Pad 23 drive strentgh<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 25 | PAD23INPEN | 0x0 | RW | Pad 23 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 24 | PAD23PULL | 0x0 | RW | Pad 23 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 23:21 | RSVD | 0x0 | RO | RESERVED |
| 20:19 | PAD22FNCSEL | 0x3 | RW | Pad 22 function select<br><br>UARTTX = 0x0 - Configure as the UART TX signal<br>M1nCE7 = 0x1 - Configure as the SPI channel 7 nCE signal from IOMSTR1<br>TCTA3 = 0x2 - Configure as the input/output signal from CTIMER A3<br>GPIO22 = 0x3 - Configure as GPIO22 |
| 18 | PAD22STRNG | 0x0 | RW | Pad 22 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 17 | PAD22INPEN | 0x0 | RW | Pad 22 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |

**Table 262: PADREGF Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 16 | PAD22PULL | 0x0 | RW | Pad 22 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12:11 | PAD21FNCSEL | 0x0 | RW | Pad 21 function select<br><br>SWDIO = 0x0 - Configure as the serial wire debug data signal<br>M1nCE6 = 0x1 - Configure as the SPI channel 6 nCE signal from IOMSTR1<br>TCTB2 = 0x2 - Configure as the input/output signal from CTIMER B2<br>GPIO21 = 0x3 - Configure as GPIO21 |
| 10 | PAD21STRNG | 0x0 | RW | Pad 21 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 9 | PAD21INPEN | 0x1 | RW | Pad 21 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD21PULL | 0x0 | RW | Pad 21 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7:5 | RSVD | 0x0 | RO | RESERVED |
| 4:3 | PAD20FNCSEL | 0x0 | RW | Pad 20 function select<br><br>SWDCK = 0x0 - Configure as the serial wire debug clock signal<br>M1nCE5 = 0x1 - Configure as the SPI channel 5 nCE signal from IOMSTR1<br>TCTA2 = 0x2 - Configure as the input/output signal from CTIMER A2<br>GPIO20 = 0x3 - Configure as GPIO20 |
| 2 | PAD20STRNG | 0x0 | RW | Pad 20 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 1 | PAD20INPEN | 0x1 | RW | Pad 20 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD20PULL | 0x0 | RW | Pad 20 pulldown enable<br><br>DIS = 0x0 - Pulldown disabled<br>EN = 0x1 - Pulldown enabled |

### *6.6.2.7 PADREGG Register*

**Pad Configuration Register G**

**OFFSET:** 0x00000018

**INSTANCE 0 ADDRESS:** 0x40010018

This register controls the pad configuration controls for PAD27 through PAD24. Writes to this register must be unlocked by the PADKEY register.

**Table 263: PADREGG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | PAD27FNCSEL | | PAD27STRNG | PAD27INPEN | PAD27PULL | RSVD | | | PAD26FNCSEL | | PAD26STRNG | PAD26INPEN | PAD26PULL | RSVD | | | PAD25FNCSEL | | PAD25STRNG | PAD25INPEN | PAD25PULL | RSVD | | | PAD24FNCSEL | | PAD24STRNG | PAD24INPEN | PAD24PULL |

**Table 264: PADREGG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:29 | RSVD | 0x0 | RO | RESERVED |
| 28:27 | PAD27FNCSEL | 0x3 | RW | Pad 27 function select<br><br>M1nCE4 = 0x1 - Configure as the SPI channel 4 nCE signal from IOMSTR1<br>TCTA1 = 0x2 - Configure as the input/output signal from CTIMER A1<br>GPIO27 = 0x3 - Configure as GPIO27 |
| 26 | PAD27STRNG | 0x0 | RW | Pad 27 drive strentgh<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 25 | PAD27INPEN | 0x0 | RW | Pad 27 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 24 | PAD27PULL | 0x0 | RW | Pad 27 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 23:21 | RSVD | 0x0 | RO | RESERVED |
| 20:19 | PAD26FNCSEL | 0x3 | RW | Pad 26 function select<br><br>M0nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR0<br>TCTB0 = 0x2 - Configure as the input/output signal from CTIMER B0<br>GPIO26 = 0x3 - Configure as GPIO26 |
| 18 | PAD26STRNG | 0x0 | RW | Pad 26 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 17 | PAD26INPEN | 0x0 | RW | Pad 26 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |

**Table 264: PADREGG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 16 | PAD26PULL | 0x0 | RW | Pad 26 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12:11 | PAD25FNCSEL | 0x3 | RW | Pad 25 function select<br><br>M0nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR0<br>TCTA0 = 0x2 - Configure as the input/output signal from CTIMER A0<br>GPIO25 = 0x3 - Configure as GPIO25 |
| 10 | PAD25STRNG | 0x0 | RW | Pad 25 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 9 | PAD25INPEN | 0x0 | RW | Pad 25 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD25PULL | 0x0 | RW | Pad 25 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7:5 | RSVD | 0x0 | RO | RESERVED |
| 4:3 | PAD24FNCSEL | 0x3 | RW | Pad 24 function select<br><br>DIS = 0x0 - Pad disabled<br>M0nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR0<br>CLKOUT = 0x2 - Configure as the CLKOUT signal<br>GPIO24 = 0x3 - Configure as GPIO24 |
| 2 | PAD24STRNG | 0x0 | RW | Pad 24 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 1 | PAD24INPEN | 0x0 | RW | Pad 24 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD24PULL | 0x0 | RW | Pad 24 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |

### *6.6.2.8 PADREGH Register*

**Pad Configuration Register H**

**OFFSET:** 0x0000001C

**INSTANCE 0 ADDRESS:** 0x4001001C

This register controls the pad configuration controls for PAD31 through PAD28. Writes to this register must be unlocked by the PADKEY register.

### Table 265: PADREGH Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | PAD31FNCSEL | | PAD31STRNG | PAD31INPEN | PAD31PULL | RSVD | | | PAD30FNCSEL | | PAD30STRNG | PAD30INPEN | PAD30PULL | RSVD | | | PAD29FNCSEL | | PAD29STRNG | PAD29INPEN | PAD29PULL | RSVD | | | PAD28FNCSEL | | PAD28STRNG | PAD28INPEN | PAD28PULL |

### Table 266: PADREGH Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:29 | RSVD | 0x0 | RO | RESERVED |
| 28:27 | PAD31FNCSEL | 0x3 | RW | Pad 31 function select<br><br>ADC6 = 0x0 - Configure as the analog ADC input 6 signal<br>M0nCE4 = 0x1 - Configure as the SPI channel 4 nCE signal from IOMSTR0<br>TCTA3 = 0x2 - Configure as the input/output signal from CTIMER A3<br>GPIO31 = 0x3 - Configure as GPIO31 |
| 26 | PAD31STRNG | 0x0 | RW | Pad 31 drive strentgh<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 25 | PAD31INPEN | 0x0 | RW | Pad 31 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 24 | PAD31PULL | 0x0 | RW | Pad 31 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 23:21 | RSVD | 0x0 | RO | RESERVED |
| 20:19 | PAD30FNCSEL | 0x3 | RW | Pad 30 function select<br><br>ADC5 = 0x0 - Configure as the analog ADC input 5 signal<br>M1nCE7 = 0x1 - Configure as the SPI channel 7 nCE signal from IOMSTR1<br>TCTB2 = 0x2 - Configure as the input/output signal from CTIMER B2<br>GPIO30 = 0x3 - Configure as GPIO30 |
| 18 | PAD30STRNG | 0x0 | RW | Pad 30 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 17 | PAD30INPEN | 0x0 | RW | Pad 30 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |

Table 266: PADREGH Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 16 | PAD30PULL | 0x0 | RW | Pad 30 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12:11 | PAD29FNCSEL | 0x3 | RW | Pad 29 function select<br><br>ADC4 = 0x0 - Configure as the analog ADC input 4 signal<br>M1nCE6 = 0x1 - Configure as the SPI channel 6 nCE signal from IOMSTR1<br>TCTA2 = 0x2 - Configure as the input/output signal from CTIMER A2<br>GPIO29 = 0x3 - Configure as GPIO29 |
| 10 | PAD29STRNG | 0x0 | RW | Pad 29 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 9 | PAD29INPEN | 0x0 | RW | Pad 29 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD29PULL | 0x0 | RW | Pad 29 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7:5 | RSVD | 0x0 | RO | RESERVED |
| 4:3 | PAD28FNCSEL | 0x3 | RW | Pad 28 function select<br><br>DIS = 0x0 - Pad disabled<br>M1nCE5 = 0x1 - Configure as the SPI channel 5 nCE signal from IOMSTR1<br>TCTB1 = 0x2 - Configure as the input/output signal from CTIMER B1<br>GPIO28 = 0x3 - Configure as GPIO28 |
| 2 | PAD28STRNG | 0x0 | RW | Pad 28 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 1 | PAD28INPEN | 0x0 | RW | Pad 28 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD28PULL | 0x0 | RW | Pad 28 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |

### *6.6.2.9 PADREGI Register*

**Pad Configuration Register I**

**OFFSET:** 0x00000020

**INSTANCE 0 ADDRESS:** 0x40010020

This register controls the pad configuration controls for PAD35 through PAD32. Writes to this register must be unlocked by the PADKEY register.

### Table 267: PADREGI Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | PAD35FNC-SEL | | PAD35STRNG | PAD35INPEN | PAD35PULL | RSVD | | | PAD34FNC-SEL | | PAD34STRNG | PAD34INPEN | PAD34PULL | RSVD | | | PAD33FNC-SEL | | PAD33STRNG | PAD33INPEN | PAD33PULL | RSVD | | | PAD32FNC-SEL | | PAD32STRNG | PAD32INPEN | PAD32PULL |

### Table 268: PADREGI Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:29 | RSVD | 0x0 | RO | RESERVED |
| 28:27 | PAD35FNCSEL | 0x3 | RW | Pad 35 function select<br><br>DIS = 0x0 - Pad disabled<br>M1nCE0 = 0x1 - Configure as the SPI channel 0 nCE signal from IOMSTR1<br>UARTTX = 0x2 - Configure as the UART TX signal<br>GPIO35 = 0x3 - Configure as GPIO35 |
| 26 | PAD35STRNG | 0x0 | RW | Pad 35 drive strentgh<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 25 | PAD35INPEN | 0x0 | RW | Pad 35 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 24 | PAD35PULL | 0x0 | RW | Pad 35 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 23:21 | RSVD | 0x0 | RO | RESERVED |
| 20:19 | PAD34FNCSEL | 0x3 | RW | Pad 34 function select<br><br>CMPRF2 = 0x0 - Configure as the analog comparator reference 2 signal<br>M0nCE7 = 0x1 - Configure as the SPI channel 7 nCE signal from IOMSTR0<br>DIS = 0x2 - Pad disabled<br>GPIO34 = 0x3 - Configure as GPIO34 |
| 18 | PAD34STRNG | 0x0 | RW | Pad 34 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |

**Table 268: PADREGI Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 17 | PAD34INPEN | 0x0 | RW | Pad 34 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 16 | PAD34PULL | 0x0 | RW | Pad 34 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12:11 | PAD33FNCSEL | 0x3 | RW | Pad 33 function select<br><br>CMPRF1 = 0x0 - Configure as the analog comparator reference 1 signal<br>M0nCE6 = 0x1 - Configure as the SPI channel 6 nCE signal from IOMSTR0<br>DIS = 0x2 - Pad disabled<br>GPIO33 = 0x3 - Configure as GPIO33 |
| 10 | PAD33STRNG | 0x0 | RW | Pad 33 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 9 | PAD33INPEN | 0x0 | RW | Pad 33 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD33PULL | 0x0 | RW | Pad 33 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7:5 | RSVD | 0x0 | RO | RESERVED |
| 4:3 | PAD32FNCSEL | 0x3 | RW | Pad 32 function select<br><br>ADC7 = 0x0 - Configure as the analog ADC input 7 signal<br>M0nCE5 = 0x1 - Configure as the SPI channel 5 nCE signal from IOMSTR0<br>TCTB3 = 0x2 - Configure as the input/output signal from CTIMER B3<br>GPIO32 = 0x3 - Configure as GPIO32 |
| 2 | PAD32STRNG | 0x0 | RW | Pad 32 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 1 | PAD32INPEN | 0x0 | RW | Pad 32 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD32PULL | 0x0 | RW | Pad 32 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |

### *6.6.2.10 PADREGJ Register*

**Pad Configuration Register J**

**OFFSET:** 0x00000024

**INSTANCE 0 ADDRESS:** 0x40010024

This register controls the pad configuration controls for PAD39 through PAD36. Writes to this register must be unlocked by the PADKEY register.

### Table 269: PADREGJ Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | PAD39FNCSEL | | PAD39STRNG | PAD39INPEN | PAD39PULL | RSVD | | | PAD38FNCSEL | | PAD38STRNG | PAD38INPEN | PAD38PULL | RSVD | | | PAD37FNCSEL | | PAD37STRNG | PAD37INPEN | PAD37PULL | RSVD | | | PAD36FNCSEL | | PAD36STRNG | PAD36INPEN | PAD36PULL |

### Table 270: PADREGJ Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:29 | RSVD | 0x0 | RO | RESERVED |
| 28:27 | PAD39FNCSEL | 0x3 | RW | Pad 39 function select<br><br>TRIG2 = 0x0 - Configure as the ADC Trigger 2 signal<br>UARTTX = 0x1 - Configure as the UART TX signal<br>CLKOUT = 0x2 - Configure as the CLKOUT signal<br>GPIO39 = 0x3 - Configure as GPIO39 |
| 26 | PAD39STRNG | 0x0 | RW | Pad 39 drive strentgh<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 25 | PAD39INPEN | 0x0 | RW | Pad 39 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 24 | PAD39PULL | 0x0 | RW | Pad 39 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 23:21 | RSVD | 0x0 | RO | RESERVED |
| 20:19 | PAD38FNCSEL | 0x3 | RW | Pad 38 function select<br><br>TRIG1 = 0x0 - Configure as the ADC Trigger 1 signal<br>M1nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR1<br>UACTS = 0x2 - Configure as the UART CTS signal<br>GPIO38 = 0x3 - Configure as GPIO38 |
| 18 | PAD38STRNG | 0x0 | RW | Pad 38 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |

Table 270: PADREGJ Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 17 | PAD38INPEN | 0x0 | RW | Pad 38 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 16 | PAD38PULL | 0x0 | RW | Pad 38 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12:11 | PAD37FNCSEL | 0x3 | RW | Pad 37 function select<br><br>TRIG0 = 0x0 - Configure as the ADC Trigger 0 signal<br>M1nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR1<br>UARTS = 0x2 - Configure as the UART RTS signal<br>GPIO37 = 0x3 - Configure as GPIO37 |
| 10 | PAD37STRNG | 0x0 | RW | Pad 37 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 9 | PAD37INPEN | 0x0 | RW | Pad 37 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD37PULL | 0x0 | RW | Pad 37 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7:5 | RSVD | 0x0 | RO | RESERVED |
| 4:3 | PAD36FNCSEL | 0x3 | RW | Pad 36 function select<br><br>DIS = 0x0 - Pad disabled<br>M1nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR1<br>UARTRX = 0x2 - Configure as the UART RX signal<br>GPIO36 = 0x3 - Configure as GPIO36 |
| 2 | PAD36STRNG | 0x0 | RW | Pad 36 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 1 | PAD36INPEN | 0x0 | RW | Pad 36 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD36PULL | 0x0 | RW | Pad 36 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |

### *6.6.2.11 PADREGK Register*

**Pad Configuration Register K**

**OFFSET:** 0x00000028

**INSTANCE 0 ADDRESS:** 0x40010028

This register controls the pad configuration controls for PAD43 through PAD40. Writes to this register must be unlocked by the PADKEY register.

**Table 271: PADREGK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | PAD43FNCSEL | | PAD43STRNG | PAD43INPEN | PAD43PULL | RSVD | | | PAD42FNCSEL | | PAD42STRNG | PAD42INPEN | PAD42PULL | RSVD | | | PAD41FNCSEL | | PAD41STRNG | PAD41INPEN | PAD41PULL | RSVD | | | PAD40FNCSEL | | PAD40STRNG | PAD40INPEN | PAD40PULL |

**Table 272: PADREGK Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:29 | RSVD | 0x0 | RO | RESERVED |
| 28:27 | PAD43FNCSEL | 0x3 | RW | Pad 43 function select<br>TRIG6 = 0x0 - Configure as the ADC Trigger 6 signal<br>M0nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR0<br>TCTB0 = 0x2 - Configure as the input/output signal from CTIMER B0<br>GPIO43 = 0x3 - Configure as GPIO43 |
| 26 | PAD43STRNG | 0x0 | RW | Pad 43 drive strentgh<br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 25 | PAD43INPEN | 0x0 | RW | Pad 43 input enable<br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 24 | PAD43PULL | 0x0 | RW | Pad 43 pullup enable<br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 23:21 | RSVD | 0x0 | RO | RESERVED |
| 20:19 | PAD42FNCSEL | 0x3 | RW | Pad 42 function select<br>TRIG5 = 0x0 - Configure as the ADC Trigger 5 signal<br>M0nCE0 = 0x1 - Configure as the SPI channel 0 nCE signal from IOMSTR0<br>TCTA0 = 0x2 - Configure as the input/output signal from CTIMER A0<br>GPIO42 = 0x3 - Configure as GPIO42 |
| 18 | PAD42STRNG | 0x0 | RW | Pad 42 drive strength<br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |

**Table 272: PADREGK Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 17 | PAD42INPEN | 0x0 | RW | Pad 42 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 16 | PAD42PULL | 0x0 | RW | Pad 42 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12:11 | PAD41FNCSEL | 0x3 | RW | Pad 41 function select<br><br>TRIG4 = 0x0 - Configure as the ADC Trigger 4 signal<br>DIS = 0x1 - Pad disabled<br>SWO = 0x2 - Configure as the serial wire debug SWO signal<br>GPIO41 = 0x3 - Configure as GPIO41 |
| 10 | PAD41STRNG | 0x0 | RW | Pad 41 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 9 | PAD41INPEN | 0x0 | RW | Pad 41 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD41PULL | 0x0 | RW | Pad 41 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7:5 | RSVD | 0x0 | RO | RESERVED |
| 4:3 | PAD40FNCSEL | 0x3 | RW | Pad 40 function select<br><br>TRIG3 = 0x0 - Configure as the ADC Trigger 3 signal<br>UARTRX = 0x1 - Configure as the UART RX signal<br>DIS = 0x2 - Pad disabled<br>GPIO40 = 0x3 - Configure as GPIO40 |
| 2 | PAD40STRNG | 0x0 | RW | Pad 40 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 1 | PAD40INPEN | 0x0 | RW | Pad 40 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD40PULL | 0x0 | RW | Pad 40 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |

### *6.6.2.12 PADREGL Register*

**Pad Configuration Register L**

**OFFSET:** 0x0000002C

**INSTANCE 0 ADDRESS:** 0x4001002C

This register controls the pad configuration controls for PAD47 through PAD44. Writes to this register must be unlocked by the PADKEY register.

**Table 273: PADREGL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | PAD47FNC-SEL | | PAD47STRNG | PAD47INPEN | PAD47PULL | RSVD | | | PAD46FNC-SEL | | PAD46STRNG | PAD46INPEN | PAD46PULL | RSVD | | | PAD45FNC-SEL | | PAD45STRNG | PAD45INPEN | PAD45PULL | RSVD | | | PAD44FNC-SEL | | PAD44STRNG | PAD44INPEN | PAD44PULL |

**Table 274: PADREGL Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:29 | RSVD | 0x0 | RO | RESERVED |
| 28:27 | PAD47FNCSEL | 0x3 | RW | Pad 47 function select<br><br>DIS = 0x0 - Pad disabled<br>M0nCE5 = 0x1 - Configure as the SPI channel 5 nCE signal from IOMSTR0<br>TCTB2 = 0x2 - Configure as the input/output signal from CTIMER B2<br>GPIO47 = 0x3 - Configure as GPIO47 |
| 26 | PAD47STRNG | 0x0 | RW | Pad 47 drive strentgh<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 25 | PAD47INPEN | 0x0 | RW | Pad 47 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 24 | PAD47PULL | 0x0 | RW | Pad 47 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 23:21 | RSVD | 0x0 | RO | RESERVED |
| 20:19 | PAD46FNCSEL | 0x3 | RW | Pad 46 function select<br><br>DIS = 0x0 - Pad disabled<br>M0nCE4 = 0x1 - Configure as the SPI channel 4 nCE signal from IOMSTR0<br>TCTA2 = 0x2 - Configure as the input/output signal from CTIMER A2<br>GPIO46 = 0x3 - Configure as GPIO46 |
| 18 | PAD46STRNG | 0x0 | RW | Pad 46 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |

**Table 274: PADREGL Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 17 | PAD46INPEN | 0x0 | RW | Pad 46 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 16 | PAD46PULL | 0x0 | RW | Pad 46 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12:11 | PAD45FNCSEL | 0x3 | RW | Pad 45 function select<br><br>DIS = 0x0 - Pad disabled<br>M0nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR0<br>TCTB1 = 0x2 - Configure as the input/output signal from CTIMER B1<br>GPIO45 = 0x3 - Configure as GPIO45 |
| 10 | PAD45STRNG | 0x0 | RW | Pad 45 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 9 | PAD45INPEN | 0x0 | RW | Pad 45 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD45PULL | 0x0 | RW | Pad 45 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7:5 | RSVD | 0x0 | RO | RESERVED |
| 4:3 | PAD44FNCSEL | 0x3 | RW | Pad 44 function select<br><br>TRIG7 = 0x0 - Configure as the ADC Trigger 7 signal<br>M0nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR0<br>TCTA1 = 0x2 - Configure as the input/output signal from CTIMER A1<br>GPIO44 = 0x3 - Configure as GPIO44 |
| 2 | PAD44STRNG | 0x0 | RW | Pad 44 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 1 | PAD44INPEN | 0x0 | RW | Pad 44 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD44PULL | 0x0 | RW | Pad 44 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |

### *6.6.2.13 PADREGM Register*

**Pad Configuration Register M**

**OFFSET:** 0x00000030

**INSTANCE 0 ADDRESS:** 0x40010030

This register controls the pad configuration controls for PAD49 through PAD48. Writes to this register must be unlocked by the PADKEY register.

**Table 275: PADREGM Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | PAD49FNCSEL | | PAD49STRNG | PAD49INPEN | PAD49PULL | RSVD | | | PAD48FNCSEL | | PAD48STRNG | PAD48INPEN | PAD48PULL |

**Table 276: PADREGM Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:13 | RSVD | 0x0 | RO | RESERVED |
| 12:11 | PAD49FNCSEL | 0x3 | RW | Pad 49 function select<br><br>DIS = 0x0 - Pad disabled<br>M0nCE7 = 0x1 - Configure as the SPI channel 7 nCE signal from IOMSTR0<br>TCTB3 = 0x2 - Configure as the input/output signal from CTIMER B3<br>GPIO49 = 0x3 - Configure as GPIO49 |
| 10 | PAD49STRNG | 0x0 | RW | Pad 49 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |
| 9 | PAD49INPEN | 0x0 | RW | Pad 49 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 8 | PAD49PULL | 0x0 | RW | Pad 49 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |
| 7:5 | RSVD | 0x0 | RO | RESERVED |
| 4:3 | PAD48FNCSEL | 0x3 | RW | Pad 48 function select<br><br>DIS = 0x0 - Pad disabled<br>M0nCE6 = 0x1 - Configure as the SPI channel 6 nCE signal from IOMSTR0<br>TCTA3 = 0x2 - Configure as the input/output signal from CTIMER A3<br>GPIO48 = 0x3 - Configure as GPIO48 |
| 2 | PAD48STRNG | 0x0 | RW | Pad 48 drive strength<br><br>LOW = 0x0 - Low drive strength<br>HIGH = 0x1 - High drive strength |

Table 276: PADREGM Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 1 | PAD48INPEN | 0x0 | RW | Pad 48 input enable<br><br>DIS = 0x0 - Pad input disabled<br>EN = 0x1 - Pad input enabled |
| 0 | PAD48PULL | 0x0 | RW | Pad 48 pullup enable<br><br>DIS = 0x0 - Pullup disabled<br>EN = 0x1 - Pullup enabled |

### 6.6.2.14 CFGA Register

**GPIO Configuration Register A**

**OFFSET:** 0x00000040

**INSTANCE 0 ADDRESS:** 0x40010040

GPIO configuration controls for GPIO[7:0]. Writes to this register must be unlocked by the PADKEY register.

Table 277: CFGA Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO7INTD | GPIO7OUTCFG | | GPIO7INCFG | GPIO6INTD | GPIO6OUTCFG | | GPIO6INCFG | GPIO5INTD | GPIO5OUTCFG | | GPIO5INCFG | GPIO4INTD | GPIO4OUTCFG | | GPIO4INCFG | GPIO3INTD | GPIO3OUTCFG | | GPIO3INCFG | GPIO2INTD | GPIO2OUTCFG | | GPIO2INCFG | GPIO1INTD | GPIO1OUTCFG | | GPIO1INCFG | GPIO0INTD | GPIO0OUTCFG | | GPIO0INCFG |

Table 278: CFGA Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | GPIO7INTD | 0x0 | RW | GPIO7 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 30:29 | GPIO7OUTCFG | 0x0 | RW | GPIO7 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 28 | GPIO7INCFG | 0x0 | RW | GPIO7 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |

**Table 278: CFGA Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 27 | GPIO6INTD | 0x0 | RW | GPIO6 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 26:25 | GPIO6OUTCFG | 0x0 | RW | GPIO6 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 24 | GPIO6INCFG | 0x0 | RW | GPIO6 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 23 | GPIO5INTD | 0x0 | RW | GPIO5 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 22:21 | GPIO5OUTCFG | 0x0 | RW | GPIO5 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 20 | GPIO5INCFG | 0x0 | RW | GPIO5 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 19 | GPIO4INTD | 0x0 | RW | GPIO4 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 18:17 | GPIO4OUTCFG | 0x0 | RW | GPIO4 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 16 | GPIO4INCFG | 0x0 | RW | GPIO4 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 15 | GPIO3INTD | 0x0 | RW | GPIO3 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 14:13 | GPIO3OUTCFG | 0x0 | RW | GPIO3 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |

### Table 278: CFGA Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 12 | GPIO3INCFG | 0x0 | RW | GPIO3 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 11 | GPIO2INTD | 0x0 | RW | GPIO2 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 10:9 | GPIO2OUTCFG | 0x0 | RW | GPIO2 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 8 | GPIO2INCFG | 0x0 | RW | GPIO2 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 7 | GPIO1INTD | 0x0 | RW | GPIO1 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 6:5 | GPIO1OUTCFG | 0x0 | RW | GPIO1 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 4 | GPIO1INCFG | 0x0 | RW | GPIO1 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 3 | GPIO0INTD | 0x0 | RW | GPIO0 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 2:1 | GPIO0OUTCFG | 0x0 | RW | GPIO0 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 0 | GPIO0INCFG | 0x0 | RW | GPIO0 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |

#### *6.6.2.15 CFGB Register*

**GPIO Configuration Register B**

**OFFSET:** 0x00000044

**INSTANCE 0 ADDRESS:** 0x40010044

GPIO configuration controls for GPIO[15:8]. Writes to this register must be unlocked by the PADKEY register.

**Table 279: CFGB Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO15INTD | GPIO15OUTCFG | | GPIO15INCFG | GPIO14INTD | GPIO14OUTCFG | | GPIO14INCFG | GPIO13INTD | GPIO13OUTCFG | | GPIO13INCFG | GPIO12INTD | GPIO12OUTCFG | | GPIO12INCFG | GPIO11INTD | GPIO11OUTCFG | | GPIO11INCFG | GPIO10INTD | GPIO10OUTCFG | | GPIO10INCFG | GPIO9INTD | GPIO9OUTCFG | | GPIO9INCFG | GPIO8INTD | GPIO8OUTCFG | | GPIO8INCFG |

**Table 280: CFGB Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | GPIO15INTD | 0x0 | RW | GPIO15 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 30:29 | GPIO15OUT-CFG | 0x0 | RW | GPIO15 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 28 | GPIO15INCFG | 0x0 | RW | GPIO15 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 27 | GPIO14INTD | 0x0 | RW | GPIO14 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 26:25 | GPIO14OUT-CFG | 0x0 | RW | GPIO14 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 24 | GPIO14INCFG | 0x0 | RW | GPIO14 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 23 | GPIO13INTD | 0x0 | RW | GPIO13 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |

**Table 280: CFGB Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 22:21 | GPIO13OUT-CFG | 0x0 | RW | GPIO13 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 20 | GPIO13INCFG | 0x0 | RW | GPIO13 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 19 | GPIO12INTD | 0x0 | RW | GPIO12 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 18:17 | GPIO12OUT-CFG | 0x0 | RW | GPIO12 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 16 | GPIO12INCFG | 0x0 | RW | GPIO12 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 15 | GPIO11INTD | 0x0 | RW | GPIO11 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 14:13 | GPIO11OUT-CFG | 0x0 | RW | GPIO11 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 12 | GPIO11INCFG | 0x0 | RW | GPIO11 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 11 | GPIO10INTD | 0x0 | RW | GPIO10 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 10:9 | GPIO10OUT-CFG | 0x0 | RW | GPIO10 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 8 | GPIO10INCFG | 0x0 | RW | GPIO10 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |

**Table 280: CFGB Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 7 | GPIO9INTD | 0x0 | RW | GPIO9 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 6:5 | GPIO9OUTCFG | 0x0 | RW | GPIO9 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 4 | GPIO9INCFG | 0x0 | RW | GPIO9 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 3 | GPIO8INTD | 0x0 | RW | GPIO8 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 2:1 | GPIO8OUTCFG | 0x0 | RW | GPIO8 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 0 | GPIO8INCFG | 0x0 | RW | GPIO8 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |

### 6.6.2.16 *CFGC Register*

**GPIO Configuration Register C**

**OFFSET:** 0x00000048

**INSTANCE 0 ADDRESS:** 0x40010048

GPIO configuration controls for GPIO[23:16]. Writes to this register must be unlocked by the PADKEY register.

**Table 281: CFGC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO23INTD | GPIO23OUTCFG | | GPIO23INCFG | GPIO22INTD | GPIO22OUTCFG | | GPIO22INCFG | GPIO21INTD | GPIO21OUTCFG | | GPIO21INCFG | GPIO20INTD | GPIO20OUTCFG | | GPIO20INCFG | GPIO19INTD | GPIO19OUTCFG | | GPIO19INCFG | GPIO18INTD | GPIO18OUTCFG | | GPIO18INCFG | GPIO17INTD | GPIO17OUTCFG | | GPIO17INCFG | GPIO16INTD | GPIO16OUTCFG | | GPIO16INCFG |

**Table 282: CFGC Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | GPIO23INTD | 0x0 | RW | GPIO23 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 30:29 | GPIO23OUT-CFG | 0x0 | RW | GPIO23 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 28 | GPIO23INCFG | 0x0 | RW | GPIO23 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 27 | GPIO22INTD | 0x0 | RW | GPIO22 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 26:25 | GPIO22OUT-CFG | 0x0 | RW | GPIO22 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 24 | GPIO22INCFG | 0x0 | RW | GPIO22 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 23 | GPIO21INTD | 0x0 | RW | GPIO21 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 22:21 | GPIO21OUT-CFG | 0x0 | RW | GPIO21 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 20 | GPIO21INCFG | 0x1 | RW | GPIO21 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 19 | GPIO20INTD | 0x0 | RW | GPIO20 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 18:17 | GPIO20OUT-CFG | 0x0 | RW | GPIO20 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |

### Table 282: CFGC Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 16 | GPIO20INCFG | 0x1 | RW | GPIO20 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 15 | GPIO19INTD | 0x0 | RW | GPIO19 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 14:13 | GPIO19OUT-CFG | 0x0 | RW | GPIO19 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 12 | GPIO19INCFG | 0x0 | RW | GPIO19 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 11 | GPIO18INTD | 0x0 | RW | GPIO18 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 10:9 | GPIO18OUT-CFG | 0x0 | RW | GPIO18 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 8 | GPIO18INCFG | 0x0 | RW | GPIO18 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 7 | GPIO17INTD | 0x0 | RW | GPIO17 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 6:5 | GPIO17OUT-CFG | 0x0 | RW | GPIO17 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 4 | GPIO17INCFG | 0x0 | RW | GPIO17 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 3 | GPIO16INTD | 0x0 | RW | GPIO16 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |

**Table 282: CFGC Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 2:1 | GPIO16OUT-CFG | 0x0 | RW | GPIO16 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 0 | GPIO16INCFG | 0x0 | RW | GPIO16 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |

### *6.6.2.17 CFGD Register*

**GPIO Configuration Register D**

**OFFSET:** 0x0000004C

**INSTANCE 0 ADDRESS:** 0x4001004C

GPIO configuration controls for GPIO[31:24]. Writes to this register must be unlocked by the PADKEY register.

**Table 283: CFGD Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO31INTD | GPIO31OUTCFG | | GPIO31INCFG | GPIO30INTD | GPIO30OUTCFG | | GPIO30INCFG | GPIO29INTD | GPIO29OUTCFG | | GPIO29INCFG | GPIO28INTD | GPIO28OUTCFG | | GPIO28INCFG | GPIO27INTD | GPIO27OUTCFG | | GPIO27INCFG | GPIO26INTD | GPIO26OUTCFG | | GPIO26INCFG | GPIO25INTD | GPIO25OUTCFG | | GPIO25INCFG | GPIO24INTD | GPIO24OUTCFG | | GPIO24INCFG |

**Table 284: CFGD Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | GPIO31INTD | 0x0 | RW | GPIO31 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 30:29 | GPIO31OUT-CFG | 0x0 | RW | GPIO31 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 28 | GPIO31INCFG | 0x0 | RW | GPIO31 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |

### Table 284: CFGD Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 27 | GPIO30INTD | 0x0 | RW | GPIO30 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 26:25 | GPIO30OUT-CFG | 0x0 | RW | GPIO30 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 24 | GPIO30INCFG | 0x0 | RW | GPIO30 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 23 | GPIO29INTD | 0x0 | RW | GPIO29 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 22:21 | GPIO29OUT-CFG | 0x0 | RW | GPIO29 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 20 | GPIO29INCFG | 0x0 | RW | GPIO29 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 19 | GPIO28INTD | 0x0 | RW | GPIO28 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 18:17 | GPIO28OUT-CFG | 0x0 | RW | GPIO28 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 16 | GPIO28INCFG | 0x0 | RW | GPIO28 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 15 | GPIO27INTD | 0x0 | RW | GPIO27 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 14:13 | GPIO27OUT-CFG | 0x0 | RW | GPIO27 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |

**Table 284: CFGD Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 12 | GPIO27INCFG | 0x0 | RW | GPIO27 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 11 | GPIO26INTD | 0x0 | RW | GPIO26 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 10:9 | GPIO26OUT-CFG | 0x0 | RW | GPIO26 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 8 | GPIO26INCFG | 0x0 | RW | GPIO26 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 7 | GPIO25INTD | 0x0 | RW | GPIO25 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 6:5 | GPIO25OUT-CFG | 0x0 | RW | GPIO25 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 4 | GPIO25INCFG | 0x0 | RW | GPIO25 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 3 | GPIO24INTD | 0x0 | RW | GPIO24 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 2:1 | GPIO24OUT-CFG | 0x0 | RW | GPIO24 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 0 | GPIO24INCFG | 0x0 | RW | GPIO24 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |

### *6.6.2.18 CFGE Register*

**GPIO Configuration Register E**

**OFFSET:** 0x00000050

**INSTANCE 0 ADDRESS:** 0x40010050

GPIO configuration controls for GPIO[39:32]. Writes to this register must be unlocked by the PADKEY register.

### Table 285: CFGE Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO39INTD | GPIO39OUTCFG | | GPIO39INCFG | GPIO38INTD | GPIO38OUTCFG | | GPIO38INCFG | GPIO37INTD | GPIO37OUTCFG | | GPIO37INCFG | GPIO36INTD | GPIO36OUTCFG | | GPIO36INCFG | GPIO35INTD | GPIO35OUTCFG | | GPIO35INCFG | GPIO34INTD | GPIO34OUTCFG | | GPIO34INCFG | GPIO33INTD | GPIO33OUTCFG | | GPIO33INCFG | GPIO32INTD | GPIO32OUTCFG | | GPIO32INCFG |

### Table 286: CFGE Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | GPIO39INTD | 0x0 | RW | GPIO39 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 30:29 | GPIO39OUTCFG | 0x0 | RW | GPIO39 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 28 | GPIO39INCFG | 0x0 | RW | GPIO39 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 27 | GPIO38INTD | 0x0 | RW | GPIO38 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 26:25 | GPIO38OUTCFG | 0x0 | RW | GPIO38 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 24 | GPIO38INCFG | 0x0 | RW | GPIO38 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 23 | GPIO37INTD | 0x0 | RW | GPIO37 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |

Table 286: CFGE Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 22:21 | GPIO37OUT-CFG | 0x0 | RW | GPIO37 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 20 | GPIO37INCFG | 0x0 | RW | GPIO37 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 19 | GPIO36INTD | 0x0 | RW | GPIO36 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 18:17 | GPIO36OUT-CFG | 0x0 | RW | GPIO36 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 16 | GPIO36INCFG | 0x0 | RW | GPIO36 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 15 | GPIO35INTD | 0x0 | RW | GPIO35 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 14:13 | GPIO35OUT-CFG | 0x0 | RW | GPIO35 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 12 | GPIO35INCFG | 0x0 | RW | GPIO35 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 11 | GPIO34INTD | 0x0 | RW | GPIO34 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 10:9 | GPIO34OUT-CFG | 0x0 | RW | GPIO34 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 8 | GPIO34INCFG | 0x0 | RW | GPIO34 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |

**Table 286: CFGE Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 7 | GPIO33INTD | 0x0 | RW | GPIO33 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 6:5 | GPIO33OUT-CFG | 0x0 | RW | GPIO33 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 4 | GPIO33INCFG | 0x0 | RW | GPIO33 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 3 | GPIO32INTD | 0x0 | RW | GPIO32 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 2:1 | GPIO32OUT-CFG | 0x0 | RW | GPIO32 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 0 | GPIO32INCFG | 0x0 | RW | GPIO32 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |

### *6.6.2.19 CFGF Register*

**GPIO Configuration Register F**

**OFFSET:** 0x00000054

**INSTANCE 0 ADDRESS:** 0x40010054

GPIO configuration controls for GPIO[47:40]. Writes to this register must be unlocked by the PADKEY register.

**Table 287: CFGF Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO47INTD | GPIO47OUTCFG | | GPIO47INCFG | GPIO46INTD | GPIO46OUTCFG | | GPIO46INCFG | GPIO45INTD | GPIO45OUTCFG | | GPIO45INCFG | GPIO44INTD | GPIO44OUTCFG | | GPIO44INCFG | GPIO43INTD | GPIO43OUTCFG | | GPIO43INCFG | GPIO42INTD | GPIO42OUTCFG | | GPIO42INCFG | GPIO41INTD | GPIO41OUTCFG | | GPIO41INCFG | GPIO40INTD | GPIO40OUTCFG | | GPIO40INCFG |

### Table 288: CFGF Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | GPIO47INTD | 0x0 | RW | GPIO47 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 30:29 | GPIO47OUT-CFG | 0x0 | RW | GPIO47 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 28 | GPIO47INCFG | 0x0 | RW | GPIO47 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 27 | GPIO46INTD | 0x0 | RW | GPIO46 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 26:25 | GPIO46OUT-CFG | 0x0 | RW | GPIO46 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 24 | GPIO46INCFG | 0x0 | RW | GPIO46 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 23 | GPIO45INTD | 0x0 | RW | GPIO45 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 22:21 | GPIO45OUT-CFG | 0x0 | RW | GPIO45 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 20 | GPIO45INCFG | 0x0 | RW | GPIO45 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 19 | GPIO44INTD | 0x0 | RW | GPIO44 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 18:17 | GPIO44OUT-CFG | 0x0 | RW | GPIO44 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |

**Table 288: CFGF Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 16 | GPIO44INCFG | 0x0 | RW | GPIO44 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 15 | GPIO43INTD | 0x0 | RW | GPIO43 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 14:13 | GPIO43OUT-CFG | 0x0 | RW | GPIO43 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 12 | GPIO43INCFG | 0x0 | RW | GPIO43 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 11 | GPIO42INTD | 0x0 | RW | GPIO42 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 10:9 | GPIO42OUT-CFG | 0x0 | RW | GPIO42 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 8 | GPIO42INCFG | 0x0 | RW | GPIO42 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 7 | GPIO41INTD | 0x0 | RW | GPIO41 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 6:5 | GPIO41OUT-CFG | 0x0 | RW | GPIO41 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 4 | GPIO41INCFG | 0x0 | RW | GPIO41 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 3 | GPIO40INTD | 0x0 | RW | GPIO40 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |

**Table 288: CFGF Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 2:1 | GPIO40OUTCFG | 0x0 | RW | GPIO40 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 0 | GPIO40INCFG | 0x0 | RW | GPIO40 input enable.<br><br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |

### *6.6.2.20 CFGG Register*

**GPIO Configuration Register G**

**OFFSET:** 0x00000058

**INSTANCE 0 ADDRESS:** 0x40010058

GPIO configuration controls for GPIO[49:48]. Writes to this register must be unlocked by the PADKEY register.

**Table 289: CFGG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | GPIO49INTD | GPIO49OUTCFG | | GPIO49INCFG | GPIO48INTD | GPIO48OUTCFG | | GPIO48INCFG |

**Table 290: CFGG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED |
| 7 | GPIO49INTD | 0x0 | RW | GPIO49 interrupt direction.<br><br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 6:5 | GPIO49OUTCFG | 0x0 | RW | GPIO49 output configuration.<br><br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |

Table 290: CFGG Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 4 | GPIO49INCFG | 0x0 | RW | GPIO49 input enable.<br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |
| 3 | GPIO48INTD | 0x0 | RW | GPIO48 interrupt direction.<br>INTLH = 0x0 - Interrupt on low to high GPIO transition<br>INTHL = 0x1 - Interrupt on high to low GPIO transition |
| 2:1 | GPIO48OUT-CFG | 0x0 | RW | GPIO48 output configuration.<br>DIS = 0x0 - Output disabled<br>PUSHPULL = 0x1 - Output is push-pull<br>OD = 0x2 - Output is open drain<br>TS = 0x3 - Output is tri-state |
| 0 | GPIO48INCFG | 0x0 | RW | GPIO48 input enable.<br>READ = 0x0 - Read the GPIO pin data<br>RDZERO = 0x1 - Readback will always be zero |

### 6.6.2.21 PADKEY Register

**Key Register for all pad configuration registers**

**OFFSET:** 0x00000060

**INSTANCE 0 ADDRESS:** 0x40010060

Key Register for all pad configuration registers

Table 291: PADKEY Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PADKEY | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 292: PADKEY Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | PADKEY | 0x0 | RW | Key register value.<br>Key = 0x73 - Key |

### 6.6.2.22 RDA Register

**GPIO Input Register A**

**OFFSET:** 0x00000080

**INSTANCE 0 ADDRESS:** 0x40010080

GPIO Input Register A

**Table 293: RDA Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RDA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 294: RDA Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | RDA | 0x0 | RO | GPIO31-0 read data. |

### 6.6.2.23 RDB Register

**GPIO Input Register B**

**OFFSET:** 0x00000084

**INSTANCE 0 ADDRESS:** 0x40010084

GPIO Input Register B

**Table 295: RDB Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | RDB | | | | | | | | | | | | | | | | | |

**Table 296: RDB Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:18 | RSVD | 0x0 | RO | RESERVED |
| 17:0 | RDB | 0x0 | RO | GPIO49-32 read data. |

### 6.6.2.24 WTA Register

**GPIO Output Register A**

**OFFSET:** 0x00000088

**INSTANCE 0 ADDRESS:** 0x40010088

GPIO Output Register A

**Table 297: WTA Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WTA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 298: WTA Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | WTA | 0x0 | RW | GPIO31-0 write data. |

### 6.6.2.25 *WTB Register*

**GPIO Output Register B**

**OFFSET:** 0x0000008C

**INSTANCE 0 ADDRESS:** 0x4001008C

GPIO Output Register B

**Table 299: WTB Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | WTB | | | | | | | | | | | | | | | | | |

**Table 300: WTB Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:18 | RSVD | 0x0 | RO | RESERVED |
| 17:0 | WTB | 0x0 | RW | GPIO49-32 write data. |

### 6.6.2.26 *WTSA Register*

**GPIO Output Register A Set**

**OFFSET:** 0x00000090

**INSTANCE 0 ADDRESS:** 0x40010090

GPIO Output Register A Set

Table 301: WTSA Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WTSA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 302: WTSA Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | WTSA | 0x0 | WO | Set the GPIO31-0 write data. |

### 6.6.2.27 *WTSB Register*

**GPIO Output Register B Set**

**OFFSET:** 0x00000094

**INSTANCE 0 ADDRESS:** 0x40010094

GPIO Output Register B Set

Table 303: WTSB Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | WTSB | | | | | | | | | | | | | | | | | |

Table 304: WTSB Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:18 | RSVD | 0x0 | RO | RESERVED |
| 17:0 | WTSB | 0x0 | WO | Set the GPIO49-32 write data. |

### 6.6.2.28 *WTCA Register*

**GPIO Output Register A Clear**

**OFFSET:** 0x00000098

**INSTANCE 0 ADDRESS:** 0x40010098

GPIO Output Register A Clear

**Table 305: WTCA Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WTCA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 306: WTCA Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | WTCA | 0x0 | WO | Clear the GPIO31-0 write data. |

### 6.6.2.29 *WTCB Register*

**GPIO Output Register B Clear**

**OFFSET:** 0x0000009C

**INSTANCE 0 ADDRESS:** 0x4001009C

GPIO Output Register B Clear

**Table 307: WTCB Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | WTCB | | | | | | | | | | | | | | | | | |

**Table 308: WTCB Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:18 | RSVD | 0x0 | RO | RESERVED |
| 17:0 | WTCB | 0x0 | WO | Clear the GPIO49-32 write data. |

### 6.6.2.30 *ENA Register*

**GPIO Enable Register A**

**OFFSET:** 0x000000A0

**INSTANCE 0 ADDRESS:** 0x400100A0

GPIO Enable Register A

**Table 309: ENA Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ENA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 310: ENA Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | ENA | 0x0 | RW | GPIO31-0 output enables |

### 6.6.2.31 ENB Register

**GPIO Enable Register B**

**OFFSET:** 0x000000A4

**INSTANCE 0 ADDRESS:** 0x400100A4

GPIO Enable Register B

**Table 311: ENB Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | ENB | | | | | | | | | | | | | | | | | |

**Table 312: ENB Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:18 | RSVD | 0x0 | RO | RESERVED |
| 17:0 | ENB | 0x0 | RW | GPIO49-32 output enables |

### 6.6.2.32 ENSA Register

**GPIO Enable Register A Set**

**OFFSET:** 0x000000A8

**INSTANCE 0 ADDRESS:** 0x400100A8

GPIO Enable Register A Set

**Table 313: ENSA Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="32">ENSA</td></tr>
</table>

**Table 314: ENSA Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | ENSA | 0x0 | RW | Set the GPIO31-0 output enables |

### 6.6.2.33 ENSB Register

**GPIO Enable Register B Set**

**OFFSET:** 0x000000AC

**INSTANCE 0 ADDRESS:** 0x400100AC

GPIO Enable Register B Set

**Table 315: ENSB Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="14">RSVD</td><td colspan="18">ENSB</td></tr>
</table>

**Table 316: ENSB Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:18 | RSVD | 0x0 | RO | RESERVED |
| 17:0 | ENSB | 0x0 | RW | Set the GPIO49-32 output enables |

### 6.6.2.34 ENCA Register

**GPIO Enable Register A Clear**

**OFFSET:** 0x000000B4

**INSTANCE 0 ADDRESS:** 0x400100B4

GPIO Enable Register A Clear

**Table 317: ENCA Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ENCA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 318: ENCA Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | ENCA | 0x0 | RW | Clear the GPIO31-0 output enables |

### 6.6.2.35 *ENCB Register*

**GPIO Enable Register B Clear**

**OFFSET:** 0x000000B8

**INSTANCE 0 ADDRESS:** 0x400100B8

GPIO Enable Register B Clear

**Table 319: ENCB Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | ENCB | | | | | | | | | | | | | | | | | |

**Table 320: ENCB Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:18 | RSVD | 0x0 | RO | RESERVED |
| 17:0 | ENCB | 0x0 | RW | Clear the GPIO49-32 output enables |

### 6.6.2.36 *INT0EN Register*

**GPIO Interrupt Registers 31-0: Enable**

**OFFSET:** 0x00000200

**INSTANCE 0 ADDRESS:** 0x40010200

Set bits in this register to allow this module to generate the corresponding interrupt.

### Table 321: INT0EN Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 | GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 | GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |

### Table 322: INT0EN Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | GPIO31 | 0x0 | RW | GPIO31 interrupt. |
| 30 | GPIO30 | 0x0 | RW | GPIO30 interrupt. |
| 29 | GPIO29 | 0x0 | RW | GPIO29 interrupt. |
| 28 | GPIO28 | 0x0 | RW | GPIO28 interrupt. |
| 27 | GPIO27 | 0x0 | RW | GPIO27 interrupt. |
| 26 | GPIO26 | 0x0 | RW | GPIO26 interrupt. |
| 25 | GPIO25 | 0x0 | RW | GPIO25 interrupt. |
| 24 | GPIO24 | 0x0 | RW | GPIO24 interrupt. |
| 23 | GPIO23 | 0x0 | RW | GPIO23 interrupt. |
| 22 | GPIO22 | 0x0 | RW | GPIO22 interrupt. |
| 21 | GPIO21 | 0x0 | RW | GPIO21 interrupt. |
| 20 | GPIO20 | 0x0 | RW | GPIO20 interrupt. |
| 19 | GPIO19 | 0x0 | RW | GPIO19 interrupt. |
| 18 | GPIO18 | 0x0 | RW | GPIO18interrupt. |
| 17 | GPIO17 | 0x0 | RW | GPIO17 interrupt. |
| 16 | GPIO16 | 0x0 | RW | GPIO16 interrupt. |
| 15 | GPIO15 | 0x0 | RW | GPIO15 interrupt. |
| 14 | GPIO14 | 0x0 | RW | GPIO14 interrupt. |

**Table 322: INT0EN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 13 | GPIO13 | 0x0 | RW | GPIO13 interrupt. |
| 12 | GPIO12 | 0x0 | RW | GPIO12 interrupt. |
| 11 | GPIO11 | 0x0 | RW | GPIO11 interrupt. |
| 10 | GPIO10 | 0x0 | RW | GPIO10 interrupt. |
| 9 | GPIO9 | 0x0 | RW | GPIO9 interrupt. |
| 8 | GPIO8 | 0x0 | RW | GPIO8 interrupt. |
| 7 | GPIO7 | 0x0 | RW | GPIO7 interrupt. |
| 6 | GPIO6 | 0x0 | RW | GPIO6 interrupt. |
| 5 | GPIO5 | 0x0 | RW | GPIO5 interrupt. |
| 4 | GPIO4 | 0x0 | RW | GPIO4 interrupt. |
| 3 | GPIO3 | 0x0 | RW | GPIO3 interrupt. |
| 2 | GPIO2 | 0x0 | RW | GPIO2 interrupt. |
| 1 | GPIO1 | 0x0 | RW | GPIO1 interrupt. |
| 0 | GPIO0 | 0x0 | RW | GPIO0 interrupt. |

### *6.6.2.37 INT0STAT Register*

**GPIO Interrupt Registers 31-0: Status**

**OFFSET:** 0x00000204

**INSTANCE 0 ADDRESS:** 0x40010204

Read bits from this register to discover the cause of a recent interrupt.

**Table 323: INT0STAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 | GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 | GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |

### Table 324: INT0STAT Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | GPIO31 | 0x0 | RW | GPIO31 interrupt. |
| 30 | GPIO30 | 0x0 | RW | GPIO30 interrupt. |
| 29 | GPIO29 | 0x0 | RW | GPIO29 interrupt. |
| 28 | GPIO28 | 0x0 | RW | GPIO28 interrupt. |
| 27 | GPIO27 | 0x0 | RW | GPIO27 interrupt. |
| 26 | GPIO26 | 0x0 | RW | GPIO26 interrupt. |
| 25 | GPIO25 | 0x0 | RW | GPIO25 interrupt. |
| 24 | GPIO24 | 0x0 | RW | GPIO24 interrupt. |
| 23 | GPIO23 | 0x0 | RW | GPIO23 interrupt. |
| 22 | GPIO22 | 0x0 | RW | GPIO22 interrupt. |
| 21 | GPIO21 | 0x0 | RW | GPIO21 interrupt. |
| 20 | GPIO20 | 0x0 | RW | GPIO20 interrupt. |
| 19 | GPIO19 | 0x0 | RW | GPIO19 interrupt. |
| 18 | GPIO18 | 0x0 | RW | GPIO18interrupt. |
| 17 | GPIO17 | 0x0 | RW | GPIO17 interrupt. |
| 16 | GPIO16 | 0x0 | RW | GPIO16 interrupt. |
| 15 | GPIO15 | 0x0 | RW | GPIO15 interrupt. |
| 14 | GPIO14 | 0x0 | RW | GPIO14 interrupt. |
| 13 | GPIO13 | 0x0 | RW | GPIO13 interrupt. |
| 12 | GPIO12 | 0x0 | RW | GPIO12 interrupt. |
| 11 | GPIO11 | 0x0 | RW | GPIO11 interrupt. |
| 10 | GPIO10 | 0x0 | RW | GPIO10 interrupt. |

**Table 324: INT0STAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 9 | GPIO9 | 0x0 | RW | GPIO9 interrupt. |
| 8 | GPIO8 | 0x0 | RW | GPIO8 interrupt. |
| 7 | GPIO7 | 0x0 | RW | GPIO7 interrupt. |
| 6 | GPIO6 | 0x0 | RW | GPIO6 interrupt. |
| 5 | GPIO5 | 0x0 | RW | GPIO5 interrupt. |
| 4 | GPIO4 | 0x0 | RW | GPIO4 interrupt. |
| 3 | GPIO3 | 0x0 | RW | GPIO3 interrupt. |
| 2 | GPIO2 | 0x0 | RW | GPIO2 interrupt. |
| 1 | GPIO1 | 0x0 | RW | GPIO1 interrupt. |
| 0 | GPIO0 | 0x0 | RW | GPIO0 interrupt. |

### *6.6.2.38 INT0CLR Register*

**GPIO Interrupt Registers 31-0: Clear**

**OFFSET:** 0x00000208

**INSTANCE 0 ADDRESS:** 0x40010208

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

**Table 325: INT0CLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 | GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 | GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |

**Table 326: INT0CLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | GPIO31 | 0x0 | RW | GPIO31 interrupt. |
| 30 | GPIO30 | 0x0 | RW | GPIO30 interrupt. |

**Table 326: INT0CLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 29 | GPIO29 | 0x0 | RW | GPIO29 interrupt. |
| 28 | GPIO28 | 0x0 | RW | GPIO28 interrupt. |
| 27 | GPIO27 | 0x0 | RW | GPIO27 interrupt. |
| 26 | GPIO26 | 0x0 | RW | GPIO26 interrupt. |
| 25 | GPIO25 | 0x0 | RW | GPIO25 interrupt. |
| 24 | GPIO24 | 0x0 | RW | GPIO24 interrupt. |
| 23 | GPIO23 | 0x0 | RW | GPIO23 interrupt. |
| 22 | GPIO22 | 0x0 | RW | GPIO22 interrupt. |
| 21 | GPIO21 | 0x0 | RW | GPIO21 interrupt. |
| 20 | GPIO20 | 0x0 | RW | GPIO20 interrupt. |
| 19 | GPIO19 | 0x0 | RW | GPIO19 interrupt. |
| 18 | GPIO18 | 0x0 | RW | GPIO18interrupt. |
| 17 | GPIO17 | 0x0 | RW | GPIO17 interrupt. |
| 16 | GPIO16 | 0x0 | RW | GPIO16 interrupt. |
| 15 | GPIO15 | 0x0 | RW | GPIO15 interrupt. |
| 14 | GPIO14 | 0x0 | RW | GPIO14 interrupt. |
| 13 | GPIO13 | 0x0 | RW | GPIO13 interrupt. |
| 12 | GPIO12 | 0x0 | RW | GPIO12 interrupt. |
| 11 | GPIO11 | 0x0 | RW | GPIO11 interrupt. |
| 10 | GPIO10 | 0x0 | RW | GPIO10 interrupt. |
| 9 | GPIO9 | 0x0 | RW | GPIO9 interrupt. |
| 8 | GPIO8 | 0x0 | RW | GPIO8 interrupt. |
| 7 | GPIO7 | 0x0 | RW | GPIO7 interrupt. |

**Table 326: INT0CLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 6 | GPIO6 | 0x0 | RW | GPIO6 interrupt. |
| 5 | GPIO5 | 0x0 | RW | GPIO5 interrupt. |
| 4 | GPIO4 | 0x0 | RW | GPIO4 interrupt. |
| 3 | GPIO3 | 0x0 | RW | GPIO3 interrupt. |
| 2 | GPIO2 | 0x0 | RW | GPIO2 interrupt. |
| 1 | GPIO1 | 0x0 | RW | GPIO1 interrupt. |
| 0 | GPIO0 | 0x0 | RW | GPIO0 interrupt. |

### *6.6.2.39 INT0SET Register*

**GPIO Interrupt Registers 31-0: Set**

**OFFSET:** 0x0000020C

**INSTANCE 0 ADDRESS:** 0x4001020C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

**Table 327: INT0SET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPIO31 | GPIO30 | GPIO29 | GPIO28 | GPIO27 | GPIO26 | GPIO25 | GPIO24 | GPIO23 | GPIO22 | GPIO21 | GPIO20 | GPIO19 | GPIO18 | GPIO17 | GPIO16 | GPIO15 | GPIO14 | GPIO13 | GPIO12 | GPIO11 | GPIO10 | GPIO9 | GPIO8 | GPIO7 | GPIO6 | GPIO5 | GPIO4 | GPIO3 | GPIO2 | GPIO1 | GPIO0 |

**Table 328: INT0SET Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | GPIO31 | 0x0 | RW | GPIO31 interrupt. |
| 30 | GPIO30 | 0x0 | RW | GPIO30 interrupt. |
| 29 | GPIO29 | 0x0 | RW | GPIO29 interrupt. |
| 28 | GPIO28 | 0x0 | RW | GPIO28 interrupt. |
| 27 | GPIO27 | 0x0 | RW | GPIO27 interrupt. |

Table 328: INT0SET Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 26 | GPIO26 | 0x0 | RW | GPIO26 interrupt. |
| 25 | GPIO25 | 0x0 | RW | GPIO25 interrupt. |
| 24 | GPIO24 | 0x0 | RW | GPIO24 interrupt. |
| 23 | GPIO23 | 0x0 | RW | GPIO23 interrupt. |
| 22 | GPIO22 | 0x0 | RW | GPIO22 interrupt. |
| 21 | GPIO21 | 0x0 | RW | GPIO21 interrupt. |
| 20 | GPIO20 | 0x0 | RW | GPIO20 interrupt. |
| 19 | GPIO19 | 0x0 | RW | GPIO19 interrupt. |
| 18 | GPIO18 | 0x0 | RW | GPIO18interrupt. |
| 17 | GPIO17 | 0x0 | RW | GPIO17 interrupt. |
| 16 | GPIO16 | 0x0 | RW | GPIO16 interrupt. |
| 15 | GPIO15 | 0x0 | RW | GPIO15 interrupt. |
| 14 | GPIO14 | 0x0 | RW | GPIO14 interrupt. |
| 13 | GPIO13 | 0x0 | RW | GPIO13 interrupt. |
| 12 | GPIO12 | 0x0 | RW | GPIO12 interrupt. |
| 11 | GPIO11 | 0x0 | RW | GPIO11 interrupt. |
| 10 | GPIO10 | 0x0 | RW | GPIO10 interrupt. |
| 9 | GPIO9 | 0x0 | RW | GPIO9 interrupt. |
| 8 | GPIO8 | 0x0 | RW | GPIO8 interrupt. |
| 7 | GPIO7 | 0x0 | RW | GPIO7 interrupt. |
| 6 | GPIO6 | 0x0 | RW | GPIO6 interrupt. |
| 5 | GPIO5 | 0x0 | RW | GPIO5 interrupt. |
| 4 | GPIO4 | 0x0 | RW | GPIO4 interrupt. |

**Table 328: INT0SET Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 3 | GPIO3 | 0x0 | RW | GPIO3 interrupt. |
| 2 | GPIO2 | 0x0 | RW | GPIO2 interrupt. |
| 1 | GPIO1 | 0x0 | RW | GPIO1 interrupt. |
| 0 | GPIO0 | 0x0 | RW | GPIO0 interrupt. |

### *6.6.2.40 INT1EN Register*

**GPIO Interrupt Registers 49-32: Enable**

**OFFSET:** 0x00000210

**INSTANCE 0 ADDRESS:** 0x40010210

Set bits in this register to allow this module to generate the corresponding interrupt.

**Table 329: INT1EN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | GPIO49 | GPIO48 | GPIO47 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 | GPIO39 | GPIO38 | GPIO37 | GPIO36 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |

**Table 330: INT1EN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:18 | RSVD | 0x0 | RO | RESERVED |
| 17 | GPIO49 | 0x0 | RW | GPIO49 interrupt. |
| 16 | GPIO48 | 0x0 | RW | GPIO48 interrupt. |
| 15 | GPIO47 | 0x0 | RW | GPIO47 interrupt. |
| 14 | GPIO46 | 0x0 | RW | GPIO46 interrupt. |
| 13 | GPIO45 | 0x0 | RW | GPIO45 interrupt. |
| 12 | GPIO44 | 0x0 | RW | GPIO44 interrupt. |
| 11 | GPIO43 | 0x0 | RW | GPIO43 interrupt. |

**Table 330: INT1EN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 10 | GPIO42 | 0x0 | RW | GPIO42 interrupt. |
| 9 | GPIO41 | 0x0 | RW | GPIO41 interrupt. |
| 8 | GPIO40 | 0x0 | RW | GPIO40 interrupt. |
| 7 | GPIO39 | 0x0 | RW | GPIO39 interrupt. |
| 6 | GPIO38 | 0x0 | RW | GPIO38 interrupt. |
| 5 | GPIO37 | 0x0 | RW | GPIO37 interrupt. |
| 4 | GPIO36 | 0x0 | RW | GPIO36 interrupt. |
| 3 | GPIO35 | 0x0 | RW | GPIO35 interrupt. |
| 2 | GPIO34 | 0x0 | RW | GPIO34 interrupt. |
| 1 | GPIO33 | 0x0 | RW | GPIO33 interrupt. |
| 0 | GPIO32 | 0x0 | RW | GPIO32 interrupt. |

### *6.6.2.41 INT1STAT Register*

**GPIO Interrupt Registers 49-32: Status**

**OFFSET:** 0x00000214

**INSTANCE 0 ADDRESS:** 0x40010214

Read bits from this register to discover the cause of a recent interrupt.

**Table 331: INT1STAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | GPIO49 | GPIO48 | GPIO47 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 | GPIO39 | GPIO38 | GPIO37 | GPIO36 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |

**Table 332: INT1STAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:18 | RSVD | 0x0 | RO | RESERVED |

Table 332: INT1STAT Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 17 | GPIO49 | 0x0 | RW | GPIO49 interrupt. |
| 16 | GPIO48 | 0x0 | RW | GPIO48 interrupt. |
| 15 | GPIO47 | 0x0 | RW | GPIO47 interrupt. |
| 14 | GPIO46 | 0x0 | RW | GPIO46 interrupt. |
| 13 | GPIO45 | 0x0 | RW | GPIO45 interrupt. |
| 12 | GPIO44 | 0x0 | RW | GPIO44 interrupt. |
| 11 | GPIO43 | 0x0 | RW | GPIO43 interrupt. |
| 10 | GPIO42 | 0x0 | RW | GPIO42 interrupt. |
| 9 | GPIO41 | 0x0 | RW | GPIO41 interrupt. |
| 8 | GPIO40 | 0x0 | RW | GPIO40 interrupt. |
| 7 | GPIO39 | 0x0 | RW | GPIO39 interrupt. |
| 6 | GPIO38 | 0x0 | RW | GPIO38 interrupt. |
| 5 | GPIO37 | 0x0 | RW | GPIO37 interrupt. |
| 4 | GPIO36 | 0x0 | RW | GPIO36 interrupt. |
| 3 | GPIO35 | 0x0 | RW | GPIO35 interrupt. |
| 2 | GPIO34 | 0x0 | RW | GPIO34 interrupt. |
| 1 | GPIO33 | 0x0 | RW | GPIO33 interrupt. |
| 0 | GPIO32 | 0x0 | RW | GPIO32 interrupt. |

### *6.6.2.42 INT1CLR Register*

**GPIO Interrupt Registers 49-32: Clear**

**OFFSET:** 0x00000218

**INSTANCE 0 ADDRESS:** 0x40010218

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

### Table 333: INT1CLR Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | GPIO49 | GPIO48 | GPIO47 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 | GPIO39 | GPIO38 | GPIO37 | GPIO36 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |

### Table 334: INT1CLR Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:18 | RSVD | 0x0 | RO | RESERVED |
| 17 | GPIO49 | 0x0 | RW | GPIO49 interrupt. |
| 16 | GPIO48 | 0x0 | RW | GPIO48 interrupt. |
| 15 | GPIO47 | 0x0 | RW | GPIO47 interrupt. |
| 14 | GPIO46 | 0x0 | RW | GPIO46 interrupt. |
| 13 | GPIO45 | 0x0 | RW | GPIO45 interrupt. |
| 12 | GPIO44 | 0x0 | RW | GPIO44 interrupt. |
| 11 | GPIO43 | 0x0 | RW | GPIO43 interrupt. |
| 10 | GPIO42 | 0x0 | RW | GPIO42 interrupt. |
| 9 | GPIO41 | 0x0 | RW | GPIO41 interrupt. |
| 8 | GPIO40 | 0x0 | RW | GPIO40 interrupt. |
| 7 | GPIO39 | 0x0 | RW | GPIO39 interrupt. |
| 6 | GPIO38 | 0x0 | RW | GPIO38 interrupt. |
| 5 | GPIO37 | 0x0 | RW | GPIO37 interrupt. |
| 4 | GPIO36 | 0x0 | RW | GPIO36 interrupt. |
| 3 | GPIO35 | 0x0 | RW | GPIO35 interrupt. |
| 2 | GPIO34 | 0x0 | RW | GPIO34 interrupt. |
| 1 | GPIO33 | 0x0 | RW | GPIO33 interrupt. |

**Table 334: INT1CLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 0 | GPIO32 | 0x0 | RW | GPIO32 interrupt. |

### *6.6.2.43 INT1SET Register*

**GPIO Interrupt Registers 49-32: Set**

**OFFSET:** 0x0000021C

**INSTANCE 0 ADDRESS:** 0x4001021C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

**Table 335: INT1SET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | GPIO49 | GPIO48 | GPIO47 | GPIO46 | GPIO45 | GPIO44 | GPIO43 | GPIO42 | GPIO41 | GPIO40 | GPIO39 | GPIO38 | GPIO37 | GPIO36 | GPIO35 | GPIO34 | GPIO33 | GPIO32 |

**Table 336: INT1SET Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:18 | RSVD | 0x0 | RO | RESERVED |
| 17 | GPIO49 | 0x0 | RW | GPIO49 interrupt. |
| 16 | GPIO48 | 0x0 | RW | GPIO48 interrupt. |
| 15 | GPIO47 | 0x0 | RW | GPIO47 interrupt. |
| 14 | GPIO46 | 0x0 | RW | GPIO46 interrupt. |
| 13 | GPIO45 | 0x0 | RW | GPIO45 interrupt. |
| 12 | GPIO44 | 0x0 | RW | GPIO44 interrupt. |
| 11 | GPIO43 | 0x0 | RW | GPIO43 interrupt. |
| 10 | GPIO42 | 0x0 | RW | GPIO42 interrupt. |
| 9 | GPIO41 | 0x0 | RW | GPIO41 interrupt. |
| 8 | GPIO40 | 0x0 | RW | GPIO40 interrupt. |

Table 336: INT1SET Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 7 | GPIO39 | 0x0 | RW | GPIO39 interrupt. |
| 6 | GPIO38 | 0x0 | RW | GPIO38 interrupt. |
| 5 | GPIO37 | 0x0 | RW | GPIO37 interrupt. |
| 4 | GPIO36 | 0x0 | RW | GPIO36 interrupt. |
| 3 | GPIO35 | 0x0 | RW | GPIO35 interrupt. |
| 2 | GPIO34 | 0x0 | RW | GPIO34 interrupt. |
| 1 | GPIO33 | 0x0 | RW | GPIO33 interrupt. |
| 0 | GPIO32 | 0x0 | RW | GPIO32 interrupt. |

# 7. Clock Generator and Real Time Clock Module

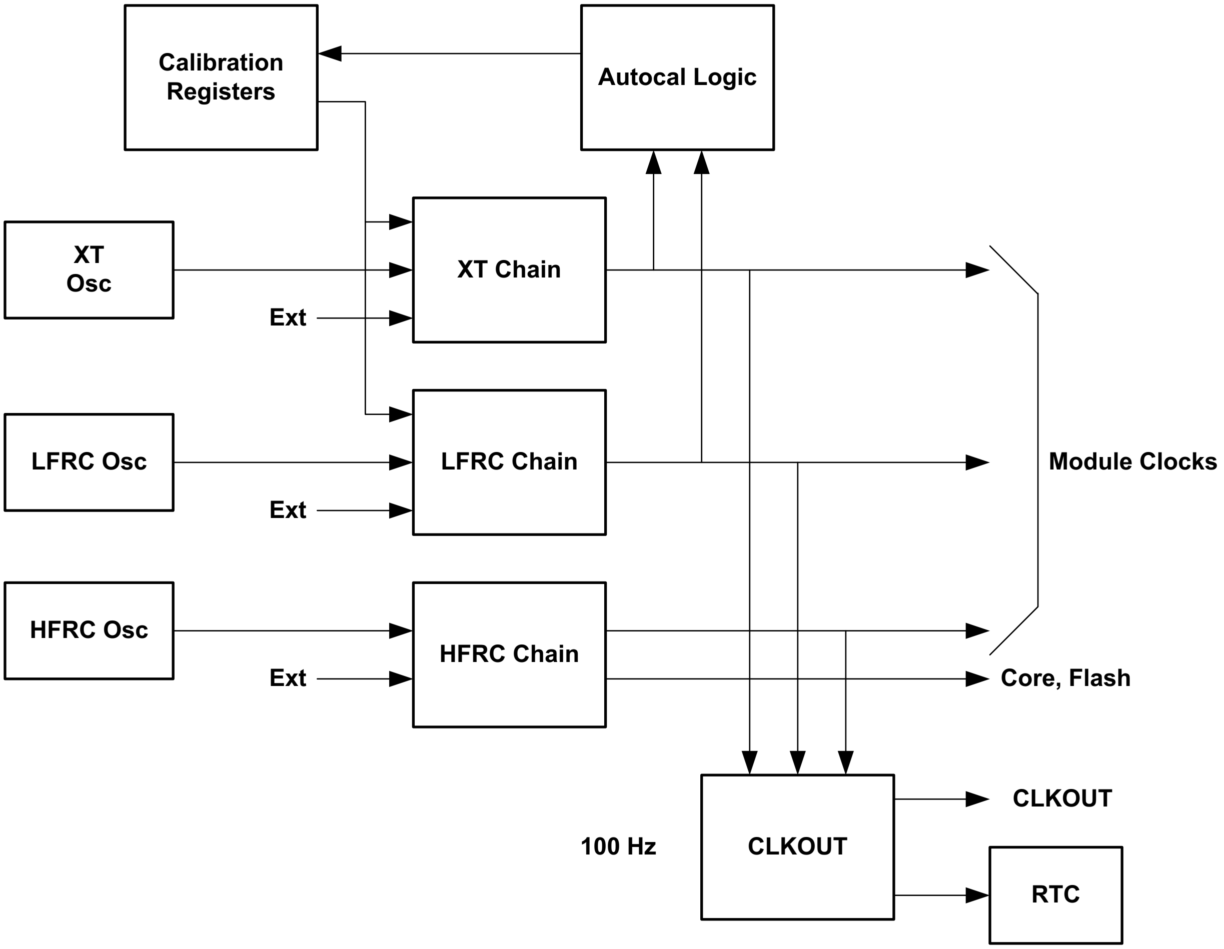


**Figure 37. Block diagram for the Clock Generator and Real Time Clock Module**

## 7.1 Clock Generator

### *7.1.1 Functional Overview*

A high-level view of the Clock Generator Module, which supplies all clocks required by the Apollo SoC, is shown in Figure 37. These clocks are derived from one of three internal oscillators: a high precision crystal controlled oscillator (XT), a low power 1 kHz RC oscillator (LFRC) and a high frequency 24 MHz oscillator (HFRC). The output of the XT oscillator may be digitally calibrated to ±1 ppm (part per million).

A clock, CLKOUT, generated from any of the oscillators, may be configured and driven onto an external pin. CLKOUT also drives the Real Time Clock (RTC) Module and other internal clock nodes.

The Clock Generator automatically controls the enabling of the oscillators, so that they are only powered up and used when requested by another module. This allows minimal power consumption without complex software intervention, so that software does not need to manage any enabling or disabling of the oscillators. As an example, an $I^2C$/SPI Master requires the HFRC in order to generate the serial interface clock. If a transfer is initiated and the processor is put into Deep Sleep mode, the HFRC will remain active until the I/O transfer is completed. At that point the HFRC will be powered down without requiring any software intervention.

### 7.1.2 Low Frequency RC Oscillator (LFRC)

The low power LFRC, with a nominal frequency of 1024 Hz, is used when short term frequency accuracy is not important. It also supplies clocks for some basic state machines and is always enabled. Calibration logic is included.

#### 7.1.2.1 LFRC Oscillator Digital Calibration

The LFRC Oscillator includes a patented Distributed Digital Calibration function similar to that of the XT Oscillator. Because the LFRC Oscillator has a greater fundamental variability, the required range of calibration is much larger. When the 1024 Hz RC oscillator is selected, the clock at the 512 Hz level of the divider chain is modified on a selectable interval using the calibration value CALRC in the CLKGEN_CALRC Register. Clock pulses are either added or subtracted to ensure accuracy of the LFRC. CALRC cycles of the 512 Hz clock are gated (negative calibration) or replaced by 1024 Hz pulses (positive calibration) within every 1024 second calibration period. Each step in CALRC modifies the clock frequency by 1.907 ppm, with a maximum adjustment of +249,954/-249,955 ppm (±25%).

The pulses which are added to or subtracted from the 512 Hz clock are spread evenly over each 1024 second period using the Ambiq Micro patented Distributed Calibration algorithm. This ensures that in LFRC mode the maximum cycle-to-cycle jitter in any clock of a frequency 512 Hz or lower caused by calibration will be no more than one 512 Hz period (~2 ms). This maximum jitter applies to all clocks in the Apollo SoC which use the LFRC.

> NOTE
>
> Since the 512 Hz LFRC clock is calibrated, the original 1024 Hz LFRC is an uncalibrated clock. This may be a useful selection in some cases.

#### 7.1.2.2 LFRC Calibration Process

The LFRC oscillator calibration value is determined by the following process:

1. Write "0x47" to the CLKKEY register to enable access to CLKGEN registers.
2. Set the CALRC field to 0 to insure calibration is not occurring.
3. Select the LFRC oscillator by setting the CLKGEN_OCTRL_OSEL bit to 1.
4. Select the LFRC or a division of it on a CLKOUT pad.
5. Measure the frequency Fmeas at the CLKOUT pad.
6. Compute the adjustment value required in ppm as ((Fnom – Fmeas)*1000000)/Fmeas = PAdj.
7. Compute the adjustment value in steps as PAdj/(1000000/2^19) = PAdj/(1.90735) = Adj.
8. If Adj < -249,955, the LFRC frequency is too high to be calibrated.
9. Else if Adj < 249,954, set CALRC = Adj.
10. Else the LFRC frequency is too low or too high to be calibrated.

### 7.1.3 High Precision XT Oscillator (XT)

The high accuracy XT Oscillator is tuned to an external 32.768 kHz crystal, and has a nominal frequency of 32.768 kHz. It is used when short term frequency accuracy is critically important. Because a crystal oscillator uses a significant amount of power, the XT is only enabled when an internal module is using it. Digital calibration logic is included.

#### 7.1.3.1 XT Oscillator Digital Calibration

The XT Oscillator includes a Distributed Digital Calibration function. When the 32 kHz XT oscillator is selected, the clock at the 16 kHz level of the divider chain is modified on a selectable interval using the calibration value CALXT in the CLKGEN_CALXT Register. Clock pulses are either added or subtracted to ensure accuracy of the XT. CALXT cycles of the 16 kHz clock are gated (negative calibration) or replaced by 32 kHz pulses (positive calibration) within every 64 second calibration period. Each step in CALXT modifies the clock frequency by 0.9535 ppm, with a maximum adjustment of +975/-976 ppm (±0.1%).

The pulses which are added to or subtracted from the 16 kHz clock are spread evenly over each 64 second period using the Ambiq Micro patented Distributed Calibration algorithm. This insures that in XT mode the maximum cycle-to-cycle jitter in any clock of a frequency 16 kHz or lower caused by calibration will be no more than one 16 kHz period (~60 us). This maximum jitter applies to all clocks in the Apollo SoC which use the XT.

> NOTE
>
> Since the 16 kHz XT clock is calibrated, the 32 kHz XT is an uncalibrated clock. This may be a useful selection in some cases.

#### 7.1.3.2 XT Calibration Process

The XT Oscillator calibration value is determined by the following process:

1. Write "0x47" to the CLKKEY register to enable access to CLKGEN registers
2. Set the CALXT register field to 0 to insure calibration is not occurring.
3. Select the XT oscillator by setting the CLKGEN_OCTRL_OSEL bit to 0.
4. Select the XT or a division of it on a CLKOUT pad.
5. Measure the frequency Fmeas at the CLKOUT pad.
6. Compute the adjustment value required in ppm as ((Fnom – Fmeas)*1000000)/Fmeas = PAdj.
7. Compute the adjustment value in steps as PAdj/(1000000/$2^{19}$) = PAdj/(0.9535) = Adj.
8. If Adj < -976, the XT frequency is too high to be calibrated.
9. Else if Adj < 975, set CALXT = Adj.
10. Else the XT frequency is too low to be calibrated.

### 7.1.4 High Frequency RC Oscillator (HFRC)

The high frequency HFRC Oscillator, with a nominal frequency of 24 MHz, is used to supply all high frequency clocks in the Apollo SoC such as the processor clock for the ARM core. Because cycle-to-cycle jitter is very critical for the HFRC and absolute accuracy is typically not critical, there is no digital calibration function for the HFRC.

The HFRC is enabled only when it is required by an internal module. When the ARM core goes into a sleep mode, the HFRC will be disabled unless another module is using it. If the ARM core goes into deep sleep mode, the HFRC will be powered down when it is not needed. Because the HFRC may not power up with an exact frequency, internal logic gates the HFRC after a power up until it is stable.

### 7.1.5 HFRC Auto-adjustment

In some applications it is important that the HFRC frequency be more accurate than the ±2% variation typically seen, particularly in cases where the temperature may vary widely. A good example of this is in cases where the Apollo SoC communicates with another device via the UART. The frequency matching with the other device in the connection is an important factor in the reliability of the connection. In order to support a highly accurate HFRC, a function called Auto-adjustment is provided.

During auto-adjustment, the number of HFRC cycles which occur in one 32.768 kHz XT Oscillator cycle is compared to a target value. If the count is different from the target, an analog HFRC tuning value is modified to change the HFRC frequency. The target count is held in the CLKGEN_HFADJ_HFXTADJ field. If the target HFRC frequency is 24 MHz, the optimal HFXTADJ value is 24,000/32.768 or 732. A different value will result in a different nominal HFRC frequency.

Auto-adjustment works by periodically enabling the HFRC and the XT, counting the HFRC cycles in a single XT cycle, subtracting that value from HFXTADJ and adding the resulting difference to the actual HFRC tuning value. The current analog tuning value may be read back in the HFTUNERB field of the CLKGEN_HFVAL Register. Auto-adjustment is enabled in the CLKGEN_HFADJ Register by loading the repeat frequency value into the HFADJCK field and then setting the HFADJEN bit.

Auto-adjustment cycles will occur continuously if both the XT and the HFRC are enabled. If either oscillator is disabled, Auto-adjustment cycles will then occur at intervals determined by the CLKGEN_HFADJ_HFADJCK field, as shown in the register description in Section 7.2.2.9 on page 239. Shorter repeat intervals will result in more accurate HFRC frequencies, especially if the temperature is changing rapidly, but will result in higher power consumption. When an Auto-adjustment cycle occurs, if the XT was disabled it is enabled and then a delay occurs to allow the XT to stabilize. This delay is defined by the CLKGEN_HFADJ_HFWARMUP field as defined in the Register document. Once the HFRC is stable, the HFRC is enabled and several Auto-adjustments occur, each of which results in a refinement of the tuning value. Once those adjustments are complete, the HFRC and XT are powered down unless they are in use by other functions.

### *7.1.6 Frequency Measurement*

The Autocalibration logic may be used to measure the frequency of an internal clock signal relative to the XT Oscillator frequency. The following steps are required to perform this measurement:

1. Write "0x47" to the CLKKEY register to enable access to CLKGEN registers
2. Set the CLKGEN_OCTRL_ACAL field to 000.
3. Clear the ACC interrupt flag.
4. Select the clock to be measured with the CKSEL CLKGEN_CLKOUT_CKSEL field.
5. Set ACAL to 110.
6. Wait for the ACC interrupt flag to be set.
7. Read the CLKGEN_ACALCTR_ACALCTR field. This will contain the number of reference clocks which occurred during one cycle of CLKOUT.
8. Calculate the frequency of the measured clock.

The measured frequency is:

$F_{MEAS} = F_{REF} \div ACALCTR$

where $F_{REF}$ is the frequency of the reference clock and ACALCTR is the value read from ACALCTR when the measurement is complete. Note that the longer the measurement period is, the more time the measurement takes, but the resulting $F_{MEAS}$ will be more accurate.

### *7.1.7 Generating 100 Hz*

The Real Time Clock (RTC) module requires a 100 Hz clock which is provided by the Clock Generator. This clock may come either from the LFRC or the XT Oscillators, as determined by the CLKGEN_OCTRL_OSEL bit. Since 100 Hz is not a simple power of two division of either of these oscillators, special functions are used to create it.

If the XT Oscillator is selected, 100 Hz is generated by dividing the 2048 Hz division of the XT by 21 for 12 iterations and by 20 for 13 iterations out of every 25 clock periods. This produces an effective division of:

$(21 * 12 + 20 * 13)/25 = 20.48$

producing an exact average frequency of 100 Hz with a maximum jitter of less than 1 ms.

If the LFRC Oscillator is selected, 100 Hz is generated by dividing the 256 Hz division of the LFRC by 2 for 11 iterations and by 3 for 14 iterations out of every 25 clock periods. This produces an effective division of:

(2 * 11 + 3 * 14)/25 = 2.56

producing an exact average frequency of 100 Hz with a maximum jitter of less than 8 ms.

### *7.1.8 XT Oscillator Failure Detection*

If the 32 KHz XT Oscillator generates clocks at less than 8 KHz for a period of more than 32 ms, the Apollo SoC detects an Oscillator Failure. The OF flag is set when an Oscillator Failure occurs, and is also set when the Apollo SoC initially powers up. If the OFIE bit is set, the OF flag will generate an interrupt. The current status of the XT Oscillator can be read in the REG_CLKGEN_STATUS_OSCF bit, which will be a 1 if the XT Oscillator is not running at least 8 KHz. Note that OSCF will always be set if the LFRC Oscillator is currently selected by the REG_CLKGEN_OCTRL_OSEL bit.

If the FOS bit is set and the Apollo SoC RTC is currently using the XT Oscillator, it will automatically switch to the LFRC Oscillator on an Oscillator Failure. This guarantees that the RTC clock will not stop in any case. If the XT Oscillator experiences a temporary failure and subsequently restarts, the Apollo SoC will switch back to the XT Oscillator. The REG_CLKGEN_STATUS_OMODE bit indicates the currently selected oscillator, which may not match the oscillator requested by the REG_CLKGEN_OCTRL_OSEL bit if the XT Oscillator is not running.

### *7.1.9 HFRC Stability Delay*

When the HFRC is powered up, it may take a few microseconds for it to begin oscillating, and a few more microseconds before the output is completely stable. In order to prevent erroneous internal clocks from occurring, the internal clocks are gated until the HFRC is stable. This is controlled by the REG_MCU_CTRL_HFRC_HFRCDEL field in MCU_CTRL, which specifies a multiple of 16 HFRC clocks for which the internal clocks are held off. Note that the HFRCDEL value is the inverse of the actual delay, so a value of 0 produces a delay of 992 clocks. The HFRCDEL value is initialized in manufacturing and will rarely need to be modified.

## 7.2 CLKGEN Registers

**Clock Generator**

**INSTANCE 0 BASE ADDRESS:**0x40004000

### 7.2.1 Register Memory Map

Table 337: CLKGEN Register Map

| Address(s) | Register Name | Description |
|---|---|---|
| 0x40004000 | CALXT | XT Oscillator Control |
| 0x40004004 | CALRC | RC Oscillator Control |
| 0x40004008 | ACALCTR | Autocalibration Counter |
| 0x4000400C | OCTRL | Oscillator Control |
| 0x40004010 | CLKOUT | CLKOUT Frequency Select |
| 0x40004014 | CLKKEY | Key Register for Clock Control Register |
| 0x40004018 | CCTRL | HFRC Clock Control |
| 0x4000401C | STATUS | Clock Generator Status |
| 0x40004020 | HFADJ | HFRC Adjustment |
| 0x40004024 | HFVAL | HFADJ readback |
| 0x40004028 | CLOCKEN | Clock Enable Status |
| 0x4000402C | UARTEN | UART Enable |
| 0x40004100 | INTEN | CLKGEN Interrupt Register: Enable |
| 0x40004104 | INTSTAT | CLKGEN Interrupt Register: Status |
| 0x40004108 | INTCLR | CLKGEN Interrupt Register: Clear |
| 0x4000410C | INTSET | CLKGEN Interrupt Register: Set |

### 7.2.2 CLKGEN Registers

#### 7.2.2.1 CALXT Register

**XT Oscillator Control**

**OFFSET:** 0x00000000

**INSTANCE 0 ADDRESS:** 0x40004000

XT Oscillator Control

Table 338: CALXT Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | CALXT | | | | | | | | | | |

Table 339: CALXT Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:11 | RSVD | 0x0 | RO | RESERVED |

**Table 339: CALXT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 10:0 | CALXT | 0x0 | RW | XT Oscillator calibration value. |

### *7.2.2.2 CALRC Register*

**RC Oscillator Control**

**OFFSET:** 0x00000004

**INSTANCE 0 ADDRESS:** 0x40004004

RC Oscillator Control

**Table 340: CALRC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | CALRC | | | | | | | | | | | | | | | | | |

**Table 341: CALRC Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:18 | RSVD | 0x0 | RO | RESERVED |
| 17:0 | CALRC | 0x0 | RW | LFRC Oscillator calibration value. |

### *7.2.2.3 ACALCTR Register*

**Autocalibration Counter**

**OFFSET:** 0x00000008

**INSTANCE 0 ADDRESS:** 0x40004008

Autocalibration Counter

**Table 342: ACALCTR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | ACALCTR | | | | | | | | | | | | | | | | | | | | | | | |

**Table 343: ACALCTR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:24 | RSVD | 0x0 | RO | RESERVED |
| 23:0 | ACALCTR | 0x0 | RO | Autocalibration Counter result. |

#### *7.2.2.4 OCTRL Register*

**Oscillator Control**

**OFFSET:** 0x0000000C

**INSTANCE 0 ADDRESS:** 0x4000400C

Oscillator Control

**Table 344: OCTRL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | ACAL | | | OSEL | FOS | RSVD | | | | STOPRC | STOPXT |

**Table 345: OCTRL Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:11 | RSVD | 0x0 | RO | RESERVED |
| 10:8 | ACAL | 0x0 | RW | Autocalibration control<br><br>DIS = 0x0 - Disable Autocalibration<br>1024SEC = 0x2 - Autocalibrate every 1024 seconds<br>512SEC = 0x3 - Autocalibrate every 512 seconds<br>XTFREQ = 0x6 - Frequency measurement using XT<br>EXTFREQ = 0x7 - Frequency measurement using external clock |
| 7 | OSEL | 0x0 | RW | Selects the RTC oscillator (1 => LFRC, 0 => XT)<br><br>RTC_XT = 0x0 - RTC uses the XT<br>RTC_LFRC = 0x1 - RTC uses the LFRC |
| 6 | FOS | 0x0 | RW | Oscillator switch on failure function<br><br>DIS = 0x0 - Disable the oscillator switch on failure function<br>EN = 0x1 - Enable the oscillator switch on failure function |
| 5:2 | RSVD | 0x0 | RO | RESERVED |
| 1 | STOPRC | 0x0 | RW | Stop the LFRC Oscillator to the RTC<br><br>EN = 0x0 - Enable the LFRC Oscillator to drive the RTC<br>STOP = 0x1 - Stop the LFRC Oscillator when driving the RTC |
| 0 | STOPXT | 0x0 | RW | Stop the XT Oscillator to the RTC<br><br>EN = 0x0 - Enable the XT Oscillator to drive the RTC<br>STOP = 0x1 - Stop the XT Oscillator when driving the RTC |

### 7.2.2.5 *CLKOUT Register*

**Table 346: CALXT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:11 | RSVD | 0x0 | RO | RESERVED |
| 10:0 | CALXT | 0x0 | RW | XT Oscillator calibration value |

**CLKOUT Select OFFSET:** 0x00000010

**INSTANCE 0 ADDRESS:** 0x40004010

CLKOUT Frequency Select

**Table 347: CLKOUT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | CKEN | RSVD | CKSEL | | | | | |

**Table 348: CLKOUT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED |
| 7 | CKEN | 0x0 | RW | Enable the CLKOUT signal<br><br>DIS = 0x0 - Disable CLKOUT<br>EN = 0x1 - Enable CLKOUT |
| 6 | RSVD | 0x0 | RO | RESERVED |

Table 348: CLKOUT Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 5:0 | CKSEL | 0x0 | RW | CLKOUT signal select. If a value is programmed that does not match a selection below, the LFRC will be selected by default.<br><br>LFRC = 0x0 - LFRC<br>XT_DIV2 = 0x1 - XT / 2<br>XT_DIV4 = 0x2 - XT / 4<br>XT_DIV8 = 0x3 - XT / 8<br>XT_DIV16 = 0x4 - XT / 16<br>XT_DIV32 = 0x5 - XT / 32<br>RTC_1Hz = 0x10 - 1 Hz as selected in RTC<br>XT_DIV2M = 0x16 - XT / 2^21<br>XT = 0x17 - XT<br>CG_100Hz = 0x18 - 100 Hz as selected in CLKGEN<br>HFRC = 0x19 - HFRC<br>HFRC_DIV2 = 0x1A - HFRC / 2<br>HFRC_DIV4 = 0x1B - HFRC / 4<br>HFRC_DIV8 = 0x1C - HFRC / 8<br>HFRC_DIV32 = 0x1D - HFRC / 32<br>HFRC_DIV64 = 0x1E - HFRC / 64<br>HFRC_DIV128 = 0x1F - HFRC / 128<br>HFRC_DIV256 = 0x20 - HFRC / 256<br>FLASH_CLK = 0x22 - Flash Clock<br>LFRC_DIV2 = 0x23 - LFRC / 2<br>LFRC_DIV32 = 0x24 - LFRC / 32<br>LFRC_DIV512 = 0x25 - LFRC / 512<br>LFRC_DIV32K = 0x26 - LFRC / 32768<br>XT_DIV256 = 0x27 - XT / 256<br>XT_DIV8K = 0x28 - XT / 8192<br>XT_DIV64K = 0x29 - XT / 2^16<br>ULFRC_DIV16 = 0x2A - Uncal LFRC / 16<br>ULFRC_DIV128 = 0x2B - Uncal LFRC / 128<br>ULFRC_1Hz = 0x2C - Uncal LFRC / 1024<br>ULFRC_DIV4K = 0x2D - Uncal LFRC / 4096<br>ULFRC_DIV1M = 0x2E - Uncal LFRC / 2^20<br>HFRC_DIV64K = 0x2F - HFRC / 2^16<br>HFRC_DIV16M = 0x30 - HFRC / 2^24<br>LFRC_DIV2M = 0x31 - LFRC / 2^20<br>HFRCNE = 0x32 - HFRC (not autoenabled)<br>HFRCNE_DIV8 = 0x33 - HFRC / 8 (not autoenabled)<br>XTNE = 0x35 - XT (not autoenabled)<br>XTNE_DIV16 = 0x36 - XT / 16 (not autoenabled)<br>LFRCNE_DIV32 = 0x37 - LFRC / 32 (not autoenabled)<br>LFRCNE = 0x39 - LFRC (not autoenabled) - Default for undefined values |

### *7.2.2.6 CLKKEY Register*

**Key Register for Clock Control Register**

**OFFSET:** 0x00000014

**INSTANCE 0 ADDRESS:** 0x40004014

Key Register for Clock Control Register

Table 349: CLKKEY Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CLKKEY | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 350: CLKKEY Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | CLKKEY | 0x0 | RW | Key register value.<br><br>Key = 0x47 - Key |

### *7.2.2.7 CCTRL Register*

**OFFSET:** 0x00000018

**INSTANCE 0 ADDRESS:** 0x40004018

HFRC Clock Control

Table 351: CCTRL Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | MEMSEL | CORE-SEL | | |

Table 352: CCTRL Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:4 | RSVD | 0x0 | RO | RESERVED |
| 3 | MEMSEL | 0x0 | RW | Flash Clock divisor<br><br>HFRC_DIV25 = 0x0 - Flash Clock is HFRC / 25<br>HFRC_DIV45 = 0x1 - Flash Clock is HFRC / 45 |

Table 352: CCTRL Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 2:0 | CORESEL | 0x7 | RW | Core Clock divisor<br><br>HFRC = 0x0 - Core Clock is HFRC<br>HFRC_DIV2 = 0x1 - Core Clock is HFRC / 2<br>HFRC_DIV3 = 0x2 - Core Clock is HFRC / 3<br>HFRC_DIV4 = 0x3 - Core Clock is HFRC / 4<br>HFRC_DIV5 = 0x4 - Core Clock is HFRC / 5<br>HFRC_DIV6 = 0x5 - Core Clock is HFRC / 6<br>HFRC_DIV7 = 0x6 - Core Clock is HFRC / 7<br>HFRC_DIV8 = 0x7 - Core Clock is HFRC / 8 |

### *7.2.2.8 STATUS Register*

**Clock Generator Status**

**OFFSET:** 0x0000001C

**INSTANCE 0 ADDRESS:** 0x4000401C

Clock Generator Status

Table 353: STATUS Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | OSCF | OMODE |

Table 354: STATUS Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:2 | RSVD | 0x0 | RO | RESERVED |
| 1 | OSCF | 0x0 | RO | XT Oscillator is enabled but not oscillating |
| 0 | OMODE | 0x0 | RO | Current RTC oscillator (1 => LFRC, 0 => XT) |

### *7.2.2.9 HFADJ Register*

**HFRC Adjustment**

**OFFSET:** 0x00000020

**INSTANCE 0 ADDRESS:** 0x40004020

HFRC Adjustment

**Table 355: HFADJ Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | HFWARMUP | HFXTADJ | | | | | | | | | | | RSVD | | | | HFADJCK | | | HFADJEN |

**Table 356: HFADJ Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:20 | RSVD | 0x0 | RO | RESERVED |
| 19 | HFWARMUP | 0x0 | RW | XT warmup period for HFRC adjustment<br><br>1SEC = 0x0 - Autoadjust XT warmup period = 1-2 seconds<br>2SEC = 0x1 - Autoadjust XT warmup period = 2-4 seconds |
| 18:8 | HFXTADJ | 0x0 | RW | Target HFRC adjustment value. |
| 7:4 | RSVD | 0x0 | RO | RESERVED |
| 3:1 | HFADJCK | 0x0 | RW | Repeat period for HFRC adjustment<br><br>4SEC = 0x0 - Autoadjust repeat period = 4 seconds<br>16SEC = 0x1 - Autoadjust repeat period = 16 seconds<br>32SEC = 0x2 - Autoadjust repeat period = 32 seconds<br>64SEC = 0x3 - Autoadjust repeat period = 64 seconds<br>128SEC = 0x4 - Autoadjust repeat period = 128 seconds<br>256SEC = 0x5 - Autoadjust repeat period = 256 seconds<br>512SEC = 0x6 - Autoadjust repeat period = 512 seconds<br>1024SEC = 0x7 - Autoadjust repeat period = 1024 seconds |
| 0 | HFADJEN | 0x0 | RW | HFRC adjustment control<br><br>DIS = 0x0 - Disable the HFRC adjustment<br>EN = 0x1 - Enable the HFRC adjustment |

### *7.2.2.10 HFVAL Register*

**HFADJ readback**

**OFFSET:** 0x00000024

**INSTANCE 0 ADDRESS:** 0x40004024

HFADJ readback

**Table 357: HFVAL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | HFTUNERB | | | | | | | | | | |

**Table 358: HFVAL Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:11 | RSVD | 0x0 | RO | RESERVED |
| 10:0 | HFTUNERB | 0x0 | RO | Current HFTUNE value |

### *7.2.2.11 CLOCKEN Register*

**Clock Enable Status**

**OFFSET:** 0x00000028

**INSTANCE 0 ADDRESS:** 0x40004028

Clock Enable Status

**Table 359: CLOCKEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CLOCKEN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 360: CLOCKEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | CLOCKEN | 0x0 | RO | Clock enable status |

### *7.2.2.12 UARTEN Register*

**UART Enable**

**OFFSET:** 0x0000002C

**INSTANCE 0 ADDRESS:** 0x4000402C

UART Enable

**Table 361: UARTEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | UARTEN |

**Table 362: UARTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:1 | RSVD | 0x0 | RO | RESERVED |
| 0 | UARTEN | 0x0 | RW | UART system clock control<br><br>DIS = 0x0 - Disable the UART system clock<br>EN = 0x1 - Enable the UART system clock |

### *7.2.2.13 INTEN Register*

**CLKGEN Interrupt Register: Enable**

**OFFSET:** 0x00000100

**INSTANCE 0 ADDRESS:** 0x40004100

Set bits in this register to allow this module to generate the corresponding interrupt.

**Table 363: INTEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | ALM | OF | ACC | ACF |

**Table 364: INTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:4 | RSVD | 0x0 | RO | RESERVED |
| 3 | ALM | 0x0 | RW | RTC Alarm interrupt |
| 2 | OF | 0x0 | RW | XT Oscillator Fail interrupt |
| 1 | ACC | 0x0 | RW | Autocalibration Complete interrupt |

**Table 364: INTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 0 | ACF | 0x0 | RW | Autocalibration Fail interrupt |

### 7.2.2.14 *INTSTAT Register*

**CLKGEN Interrupt Register: Status**

**OFFSET:** 0x00000104

**INSTANCE 0 ADDRESS:** 0x40004104

Read bits from this register to discover the cause of a recent interrupt.

**Table 365: INTSTAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | ALM | OF | ACC | ACF |

**Table 366: INTSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:4 | RSVD | 0x0 | RO | RESERVED |
| 3 | ALM | 0x0 | RW | RTC Alarm interrupt |
| 2 | OF | 0x0 | RW | XT Oscillator Fail interrupt |
| 1 | ACC | 0x0 | RW | Autocalibration Complete interrupt |
| 0 | ACF | 0x0 | RW | Autocalibration Fail interrupt |

### 7.2.2.15 *INTCLR Register*

**CLKGEN Interrupt Register: Clear**

**OFFSET:** 0x00000108

**INSTANCE 0 ADDRESS:** 0x40004108

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

**Table 367: INTCLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | ALM | OF | ACC | ACF |

**Table 368: INTCLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:4 | RSVD | 0x0 | RO | RESERVED |
| 3 | ALM | 0x0 | RW | RTC Alarm interrupt |
| 2 | OF | 0x0 | RW | XT Oscillator Fail interrupt |
| 1 | ACC | 0x0 | RW | Autocalibration Complete interrupt |
| 0 | ACF | 0x0 | RW | Autocalibration Fail interrupt |

### *7.2.2.16 INTSET Register*

**CLKGEN Interrupt Register: Set**

**OFFSET:** 0x0000010C

**INSTANCE 0 ADDRESS:** 0x4000410C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

**Table 369: INTSET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | ALM | OF | ACC | ACF |

**Table 370: INTSET Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:4 | RSVD | 0x0 | RO | RESERVED |
| 3 | ALM | 0x0 | RW | RTC Alarm interrupt |

Table 370: INTSET Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 2 | OF | 0x0 | RW | XT Oscillator Fail interrupt |
| 1 | ACC | 0x0 | RW | Autocalibration Complete interrupt |
| 0 | ACF | 0x0 | RW | Autocalibration Fail interrupt |

## 7.3 Real Time Clock

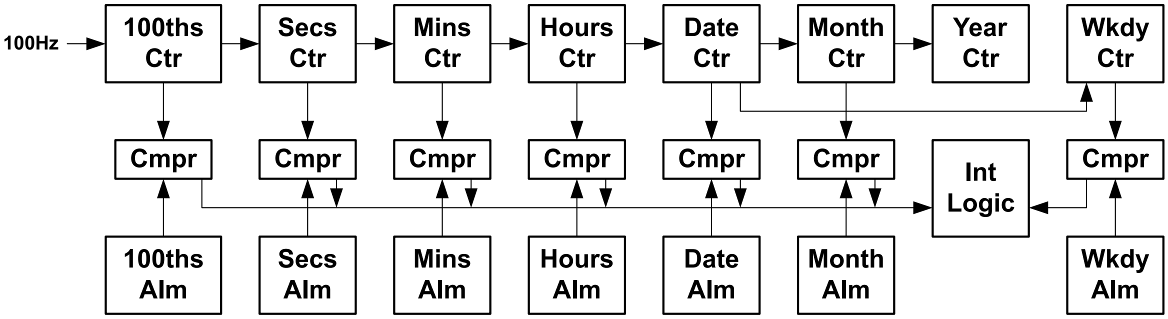


**Figure 38. Block diagram for the Real Time Clock Module**

### *7.3.1 RTC Functional Overview*

The Real Time Clock (RTC) Module, shown in Figure 38, provides an accurate real time measurement. Key features are:

- 100th of a second resolution
- Time is measured for the years between 1900 and 2199
- Automatic leap year calculation
- Hours may be specified in 12 or 24 hour mode
- Alarm precise to 1/100 second
- Alarm interval every 100th second, 10th second, second, minute, hour, day, week, month or year.
- 100 Hz input clock taken from either the high accuracy XT Oscillator or the low power LFRC Oscillator.

### *7.3.2 Calendar Counters*

The real time is held in a set of eight Calendar Counters, which hold the current 1/100th of a second (CLKGEN_CTRLOW_CTR100), the current second (CLKGEN_CTRLOW_CTRSEC), the minute (CLKGEN_CTRLOW_CTRMIN), the hour (CLKGEN_CTRLOW_CTRHR), the current day of the month (CLKGEN_CTRUP_CTRDATE), the current day of the week (CLKGEN_CTRUP_CTRWKDY), the current month (CLKGEN_CTRUP_CTRMO), the current year (CLKGEN_CTRUP_CTRYR) and the current century (CLKGEN_CTRUP_CB), all in BCD format. In order to insure that the RTC starts precisely, the timer chain which generates the 100 Hz clock is reset to 0 whenever any of the Calendar Counter Registers is written. Since unintentional modification of the Calendar Counters is a serious problem, the CLKGEN_RTCCTL_WRTC bit must be set in order to write any of the counters, and should be reset by software after any load of the Calendar Counters.

Software may stop the clock to the Calendar Counters by setting the CLKGEN_RTCCTL_RSTOP bit. This may be used in modes like Stopwatch to precisely start and stop the Calendar Counters.

### *7.3.3 Calendar Counter Reads*

The RTC includes special logic to help insure that the Calendar Counters may be read reliably, i.e. that no rollover has occurred. Because two 32-bit reads are required to read the complete set of counters, it is possible that a delay occurs between the two reads which causes a rollover to occur. An interrupt is the most likely reason this could occur. If two 100 Hz clocks occur between these two reads, the CLKGEN_CTRUP_CTRERR bit will be set. Software should check this bit after any Calendar Counter read, and perform the read again if it is set. Any read of the upper counter word will clear the CTRERR bit.

### 7.3.4 Alarms

There are seven Alarm Registers which may be used to generate an Alarm interrupt at a specific time. These registers correspond to the 100th of a second (CLKGEN_ALMLOW_ALM100), second (CLKGEN_ALMLOW_ALMSEC), minute (CLKGEN_ALMLOW_ALMMIN), hour (CLKGEN_ALMLOW_ALMHR), day of the month (CLKGEN_ALMUP_ALMDATE), day of the week (CLKGEN_ALMUP_ALMWKDY) and month (CLKGEN_ALMUP_ALMMO) Calendar Counters. The comparison is controlled by the CLKGEN_RTCCTL_RPT field and the CLKGEN_ALMLOW_ALM100 Register as shown in 12/24 Hour Mode. In the ALM100 Register, n indicates any digit 0-9. When all selected Counters match their corresponding Alarm Register, the ALM interrupt flag is set (see the Clock Generator section for the ALM interrupt control).

**Table 371: Alarm RPT Function**

| RPT Value | Interval | Comparison |
|---|---|---|
| 000 | Disabled | None |
| 001 | Every year | 100th, second, minute, hour, day, month |
| 010 | Every month | 100th, second, minute, hour, day |
| 011 | Every week | 100th, second, minute, hour, weekday |
| 100 | Every day | 100th, second, minute, hour |
| 101 | Every hour | 100th, second, minute |
| 110 | Every minute | 100th, second |
| 111 | Every second | 100th |

All alarm interrupts are asserted on the next 100 Hz clock cycle after the counters match the alarm register, except for 100ths of a second. To get an interrupt that occurs precisely at a certain time, the comparison value in the corresponding alarm register should be set 10 ms (one 100 Hz count) earlier than the desired interrupt time.

For the 100ths of a second interrupt, the first 100 Hz clock sets the comparison with the alarm register and the next clock asserts the interrupt. Therefore, the first 100ths interrupt will be asserted after 20 ms, not 10 ms. This occurs each and every time the 100ths of a second counter with interrupts is enabled if the RTC is stopped. If the RTC is already running when configured, then the first interrupt will occur between 10 and 20 ms after configuration.

### 7.3.5 12/24 Hour Mode

If the CLKGEN_RTCCTL_HR1224 bit is 0, the RTC is in 24-hour mode, and the Hours and Hours Alarm Registers hold a 6-bit BCD value which is the 24-hour time (values 0 to 23). If the HR1224 bit is 1, the RTC is in 12-hour mode, and the Hours and Hours Alarm Registers hold a 5-bit BCD value which is the 12-hour time (values 1 to 12), and bit 5 is the AP bit which is 0 for an AM time and 1 for a PM time. If the HR1224 bit is modified the Hours and Hours Alarm fields must be updated.

### 7.3.6 Century Control and Leap Year Management

The CLKGEN_CTRUP_CB bit indicates the current century. A value of 0 indicates the 20th century, and a value of 1 indicates the 19th or 21st century. The CB value will toggle when the Years counter rolls over from 99 to 0 if the CLKGEN_CTRUP_CEB bit is set, and will remain constant if CEB is clear. The century value is used to control the Leap Year functions, which create the correct insertion of February 29 in years which are divisible by 4 and not divisible by 100, and also the year 2000.

### 7.3.7 Weekday Function

The Weekday Counter is simply a 3-bit counter which counts up to 6 and then resets to 0. It is the responsibility of software to assign particular days of the week to each counter value.

## 7.4 RTC Registers

**Real Time Clock**

**INSTANCE 0 BASE ADDRESS:**0x40004000

### 7.4.1 Register Memory Map

**Table 372: RTC Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0x40004040 | CTRLOW | RTC Counters Lower |
| 0x40004044 | CTRUP | RTC Counters Upper |
| 0x40004048 | ALMLOW | RTC Alarms Lower |
| 0x4000404C | ALMUP | RTC Alarms Upper |
| 0x40004050 | RTCCTL | RTC Control Register |
| 0x40004100 | INTEN | CLK_GEN Interrupt Register: Enable |
| 0x40004104 | INTSTAT | CLK_GEN Interrupt Register: Status |
| 0x40004108 | INTCLR | CLK_GEN Interrupt Register: Clear |
| 0x4000410C | INTSET | CLK_GEN Interrupt Register: Set |

### 7.4.2 RTC Registers

#### 7.4.2.1 CTRLOW Register

**RTC Counters Lower**

**OFFSET:** 0x00000040

**INSTANCE 0 ADDRESS:** 0x40004040

RTC Counters Lower

**Table 373: CTRLOW Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | CTRHR | | | | | | RSVD | CTRMIN | | | | | | | RSVD | CTRSEC | | | | | | | CTR100 | | | | | | | |

Table 374: CTRLOW Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:30 | RSVD | 0x0 | RO | RESERVED |
| 29:24 | CTRHR | 0x1 | RW | Hours Counter |
| 23 | RSVD | 0x0 | RO | RESERVED |
| 22:16 | CTRMIN | 0x0 | RW | Minutes Counter |
| 15 | RSVD | 0x0 | RO | RESERVED |
| 14:8 | CTRSEC | 0x0 | RW | Seconds Counter |
| 7:0 | CTR100 | 0x0 | RW | 100ths of a second Counter |

### *7.4.2.2 CTRUP Register*

**RTC Counters Upper**

**OFFSET:** 0x00000044

**INSTANCE 0 ADDRESS:** 0x40004044

RTC Counters Upper

Table 375: CTRUP Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CTERR | RSVD | | CEB | CB | CTRWKDY | | | CTRYR | | | | | | | | RSVD | | | CTRMO | | | | | RSVD | | CTRDATE | | | | | |

Table 376: CTRUP Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | CTERR | 0x0 | RO | Counter read error status<br>NOERR = 0x0 - No read error occurred<br>RDERR = 0x1 - Read error occurred |
| 30:29 | RSVD | 0x0 | RO | RESERVED |

**Table 376: CTRUP Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 28 | CEB | 0x0 | RW | Century enable<br>DIS = 0x0 - Disable the Century bit from changing<br>EN = 0x1 - Enable the Century bit to change |
| 27 | CB | 0x0 | RW | Century<br>2000 = 0x0 - Century is 2000s<br>1900_2100 = 0x1 - Century is 1900s/2100s |
| 26:24 | CTRWKDY | 0x0 | RW | Weekdays Counter |
| 23:16 | CTRYR | 0x0 | RW | Years Counter |
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12:8 | CTRMO | 0x0 | RW | Months Counter |
| 7:6 | RSVD | 0x0 | RO | RESERVED |
| 5:0 | CTRDATE | 0x0 | RW | Date Counter |

### *7.4.2.3 ALMLOW Register*

**RTC Alarms Lower**

**OFFSET:** 0x00000048

**INSTANCE 0 ADDRESS:** 0x40004048

RTC Alarms Lower

**Table 377: ALMLOW Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | ALMHR | | | | | | RSVD | ALMMIN | | | | | | | RSVD | ALMSEC | | | | | | | ALM100 | | | | | | | |

**Table 378: ALMLOW Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:30 | RSVD | 0x0 | RO | RESERVED |
| 29:24 | ALMHR | 0x0 | RW | Hours Alarm |
| 23 | RSVD | 0x0 | RO | RESERVED |

**Table 378: ALMLOW Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 22:16 | ALMMIN | 0x0 | RW | Minutes Alarm |
| 15 | RSVD | 0x0 | RO | RESERVED |
| 14:8 | ALMSEC | 0x0 | RW | Seconds Alarm |
| 7:0 | ALM100 | 0x0 | RW | 100ths of a second Alarm |

### *7.4.2.4 ALMUP Register*

**RTC Alarms Upper**

**OFFSET:** 0x0000004C

**INSTANCE 0 ADDRESS:** 0x4000404C

RTC Alarms Upper

**Table 379: ALMUP Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="13">RSVD</td><td colspan="3">ALMWKDY</td><td colspan="3">RSVD</td><td colspan="5">ALMMO</td><td colspan="2">RSVD</td><td colspan="6">ALMDATE</td></tr>
</table>

**Table 380: ALMUP Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:19 | RSVD | 0x0 | RO | RESERVED |
| 18:16 | ALMWKDY | 0x0 | RW | Weekdays Alarm |
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12:8 | ALMMO | 0x0 | RW | Months Alarm |
| 7:6 | RSVD | 0x0 | RO | RESERVED |
| 5:0 | ALMDATE | 0x0 | RW | Date Alarm |

### *7.4.2.5 RTCCTL Register*

**RTC Control Register**

**OFFSET:** 0x00000050

**INSTANCE 0 ADDRESS:** 0x40004050

RTC Control Register

**Table 381: RTCCTL Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | HR1224 | RSTOP | RPT | | | WRTC |

**Table 382: RTCCTL Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:6 | RSVD | 0x0 | RO | RESERVED |
| 5 | HR1224 | 0x0 | RW | Hours Counter mode<br><br>24HR = 0x0 - Hours in 24 hour mode<br>12HR = 0x1 - Hours in 12 hour mode |
| 4 | RSTOP | 0x0 | RW | RTC input clock control<br><br>RUN = 0x0 - Allow the RTC input clock to run<br>STOP = 0x1 - Stop the RTC input clock |
| 3:1 | RPT | 0x0 | RW | Alarm repeat interval<br><br>DIS = 0x0 - Alarm interrupt disabled<br>YEAR = 0x1 - Interrupt every year<br>MONTH = 0x2 - Interrupt every month<br>WEEK = 0x3 - Interrupt every week<br>DAY = 0x4 - Interrupt every day<br>HR = 0x5 - Interrupt every hour<br>MIN = 0x6 - Interrupt every minute<br>SEC = 0x7 - Interrupt every second/10th/100th |
| 0 | WRTC | 0x0 | RW | Counter write control<br><br>DIS = 0x0 - Counter writes are disabled<br>EN = 0x1 - Counter writes are enabled |

### *7.4.2.6 INTEN Register*

**CLK_GEN Interrupt Register: Enable**

**OFFSET:** 0x00000100

**INSTANCE 0 ADDRESS:** 0x40004100

Set bits in this register to allow this module to generate the corresponding interrupt.

**Table 383: INTEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | ALM | OF | ACC | ACF |

**Table 384: INTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:4 | RSVD | 0x0 | RO | RESERVED |
| 3 | ALM | 0x0 | RW | RTC Alarm interrupt |
| 2 | OF | 0x0 | RW | XT Oscillator Fail interrupt |
| 1 | ACC | 0x0 | RW | Autocalibration Complete interrupt |
| 0 | ACF | 0x0 | RW | Autocalibration Fail interrupt |

#### *7.4.2.7 INTSTAT Register*

**CLK_GEN Interrupt Register: Status**

**OFFSET:** 0x00000104

**INSTANCE 0 ADDRESS:** 0x40004104

Read bits from this register to discover the cause of a recent interrupt.

**Table 385: INTSTAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | ALM | OF | ACC | ACF |

**Table 386: INTSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:4 | RSVD | 0x0 | RO | RESERVED |
| 3 | ALM | 0x0 | RW | RTC Alarm interrupt |

**Table 386: INTSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 2 | OF | 0x0 | RW | XT Oscillator Fail interrupt |
| 1 | ACC | 0x0 | RW | Autocalibration Complete interrupt |
| 0 | ACF | 0x0 | RW | Autocalibration Fail interrupt |

### *7.4.2.8 INTCLR Register*

**CLK_GEN Interrupt Register: Clear**

**OFFSET:** 0x00000108

**INSTANCE 0 ADDRESS:** 0x40004108

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

**Table 387: INTCLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | ALM | OF | ACC | ACF |

**Table 388: INTCLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:4 | RSVD | 0x0 | RO | RESERVED |
| 3 | ALM | 0x0 | RW | RTC Alarm interrupt |
| 2 | OF | 0x0 | RW | XT Oscillator Fail interrupt |
| 1 | ACC | 0x0 | RW | Autocalibration Complete interrupt |
| 0 | ACF | 0x0 | RW | Autocalibration Fail interrupt |

### *7.4.2.9 INTSET Register*

**CLK_GEN Interrupt Register: Set**

**OFFSET:** 0x0000010C

**INSTANCE 0 ADDRESS:** 0x4000410C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

### Table 389: INTSET Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | ALM | OF | ACC | ACF |

### Table 390: INTSET Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:4 | RSVD | 0x0 | RO | RESERVED |
| 3 | ALM | 0x0 | RW | RTC Alarm interrupt |
| 2 | OF | 0x0 | RW | XT Oscillator Fail interrupt |
| 1 | ACC | 0x0 | RW | Autocalibration Complete interrupt |
| 0 | ACF | 0x0 | RW | Autocalibration Fail interrupt |

# 8. Counter/Timer Module (CTIMER)

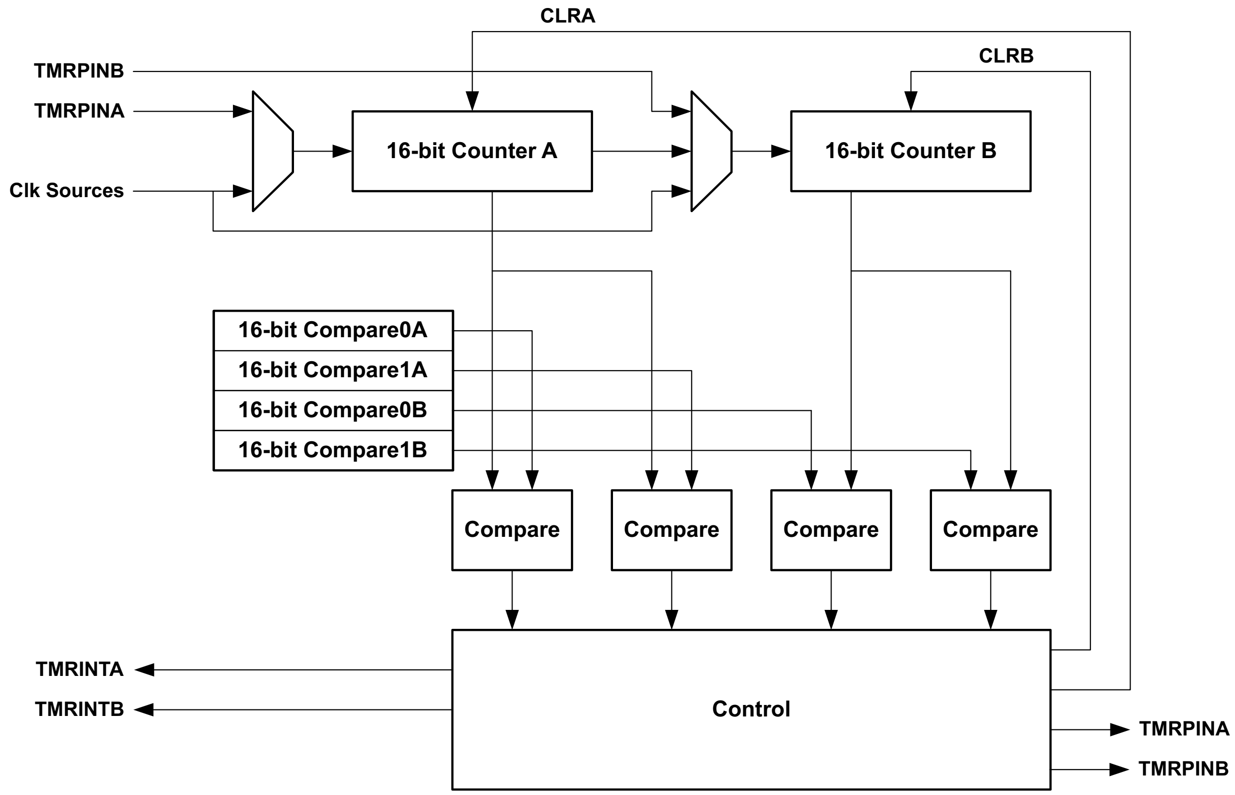


**Figure 39. Block Diagram for One Counter/Timer Pair**

## 8.1 Functional Overview

The Apollo SoC Timer/Counter module includes four Timer/Counter pairs, one of which is shown in Figure 39.. Each Timer/Counter pair includes two very low power asynchronous 16-bit counters, which may be combined to provide a 32-bit counter. Four registers contain reset values for the counters and/or comparison values to allow the generation of complex external signals. Each Timer/Counter has an external pin connection, which can be configured to provide a variety of outputs.

The features of the Timer Module are as follows:

- Interrupt after a specified delay
- Interrupt periodically with a specified period
- Determine the time between events
- Generate an external pulse of a specified width, configurable after a specified delay
- Generate an external PWM signal with a specified period and duty cycle
- Count edges on an external input
- Interrupt after a specified number of external pulses

## 8.2 Counter/Timer Functions

Each Counter/Timer operates in a mode controlled by the CTIMER_CTCTRLx_TMRxyFN bit field (x=0 to 3, y=A or B). The mode affects both the generation of interrupts and the control of an external pin. Each mode is described in the following sections. Note that for all functions, a CTIMER_CMPR0/1 value of zero (a count of 1) is invalid, and that the first measured period will be between the CTIMER_CMPR0 value plus 2 and the specified value plus 3. Subsequent repeated cycles will be correctly of length (CMPR value + 1). There are five modes:

0 => Single Count: Timer counts from 0 to CMPR0 and stops, with an optional interrupt.

1 => Repeated Count: Periodic 1-clock-cycle wide pulses with optional interrupts.

2 => Single Pulse (One Shot): Timer counts from 0 to CMPR0, generates a pin transition and an optional interrupt, continues counting from CMPR0 to CMPR1, generates another pin transition (with no interrupt), and then stops.

3 => Repeated Pulse: Same as single pulse, but the timer rolls over to 0 and restarts immediately after reaching CMPR1. Often used to generate PWM signals.

6 => Continuous: The timer repeatedly counts from 0 to $2^{16}$ - 1 forever, regardless of what the CMPR values are. The timer can optionally generate an interrupt or pin transition the first time it reaches CMPR0, but it won't generate an interrupt or pin transition on subsequent counter cycles.

### *8.2.1 Single Count (FN = 0)*

Operation in this mode is shown in Figure 40. When the Timer is enabled, the pin output is at the level selected by the POL bit and the Timer is at zero because CLR has been asserted previously. The Timer counts up on each selected clock, and when it reaches the value in the corresponding CTIMER_CMPR0 Register the output pin switches polarity (if the PE bit is set) and an interrupt is generated (if the IE bit is set). At this point the Timer resets to 0 and the output pin is maintained at the selected level until the Timer is cleared with CLR. The interrupt may be cleared by writing the corresponding WC bit in the TMRWCR Register.

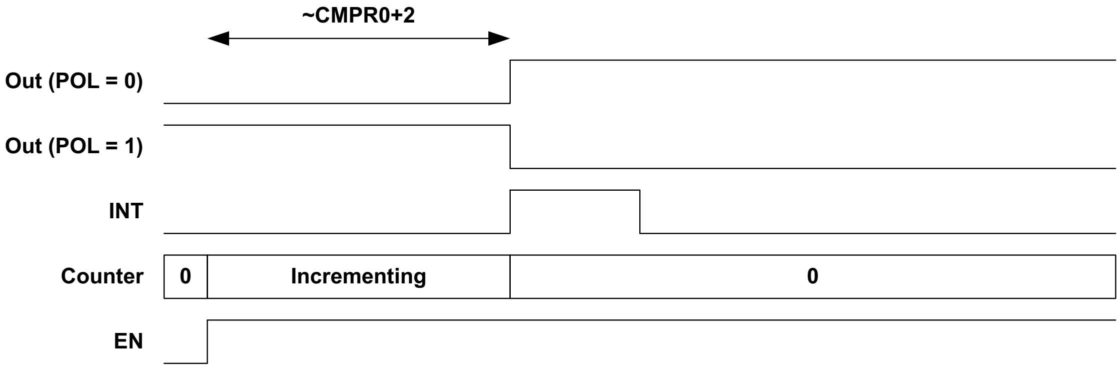


**Figure 40. Counter/Timer Operation, FN = 0**

### *8.2.2 Repeated Count (FN = 1)*

Operation in this mode is shown in Figure 41. When the Timer is enabled, the pin output is at the level selected by the POL bit and the Timer is at zero because CTIMER_TMRxyCLR has been asserted previously. The Timer counts up on each selected clock, and when it reaches the value in the corresponding CMPR0 Register the output pin switches polarity (if the CTIMER_TMRxyPE bit is set) and an interrupt is generated (if the IE bit is set). At this point the Timer resets to 0 and the output pin is maintained at the selected level for one clock cycle, after which it returns to the original value. The Timer continues to count up and the process is repeated, creating a stream of pulses or interrupts at a fixed interval. The interrupt may be cleared by writing the corresponding WC bit in the TMRWCR Register at any point prior to the next setting pulse.

If the CTIMER_TMRxyEN bit is cleared, the Timer will stop counting but will not be cleared, so the sequence may be paused and then resumed. Setting CLR will reset the Timer to zero. Note that CMPR0 must be at least 1 so that the repeat interval is two clock cycles.

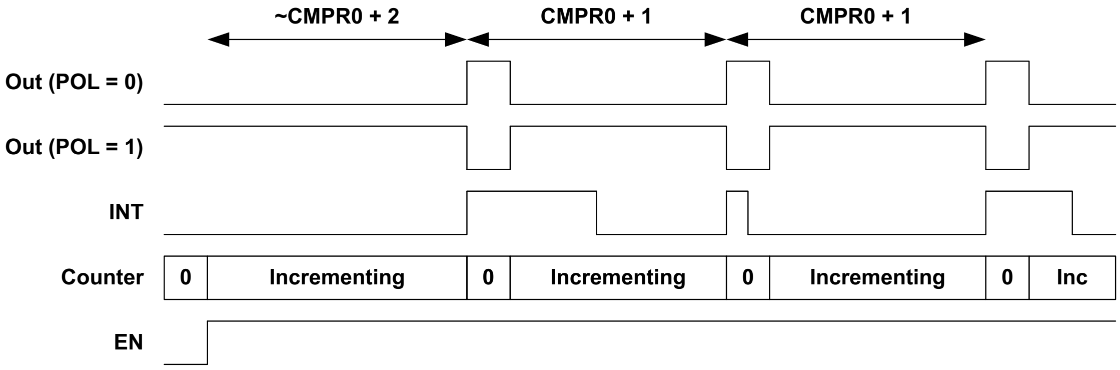


**Figure 41. Counter/Timer Operation, FN = 1**

### 8.2.3 *Single Pulse (FN = 2)*

Operation in this mode is shown in Figure 42. When the Timer is enabled, the pin output is at the level selected by the CTIMER_TMRxyPOL bit and the Timer is at zero because CLR has been asserted previously. The Timer counts up on each selected clock, and when it reaches the value in the corresponding CMPR0 Register the output pin switches polarity (if the CTIMER_TMRxyPE bit is set) and an interrupt is generated (if the CTIMER_TMRxyIE bit is set). At this point the Timer continues to increment and the output pin is maintained at the selected level until the Timer reaches the value in the CMPR1 Register, at which point it switches back to the original level. This allows the creation of a pulse of a specified width. The Timer is reset to 0 so that a single pulse is created. The interrupt may be cleared by writing the corresponding WC bit in the TMRWCR Register.

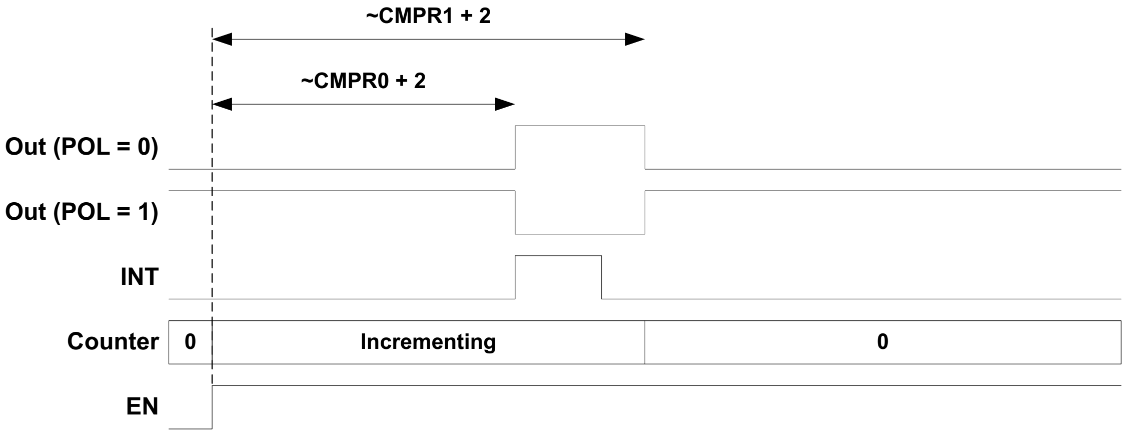


**Figure 42. Counter/Timer Operation, FN = 2**

### 8.2.4 *Repeated Pulse (FN = 3)*

Operation in this mode is shown in Figure 43. When the Timer is enabled, the pin output is at the level selected by the POL bit and the Timer is at zero because CLR has been asserted previously. The Timer counts up on each selected clock, and when it reaches the value in the corresponding CMPR0 Register the output pin switches polarity (if the PE bit is set) and an interrupt is generated (if the IE bit is set). At this point the Timer continues to increment and the output pin is maintained at the selected level until the Timer reaches the value in the CMPR1 Register, at which point it switches back to the original level. This allows the creation of a pulse of a specified width. The interrupt may be cleared by writing the corresponding WC bit in the TMRWCR Register. Note that CMPR1 must be at least 1 so that the repeat interval is two clock cycles.

The Timer is reset to 0 and continues to increment, so that a stream of pulses of the specified width and period is generated. If the EN bit is cleared, the Timer stops counting, but is not cleared, so the sequence may be paused and restarted. This mode is particularly valuable for creating a PWM (Pulse Width Modulation) output on the pin which may be used, for example, to vary the brightness of an LED.

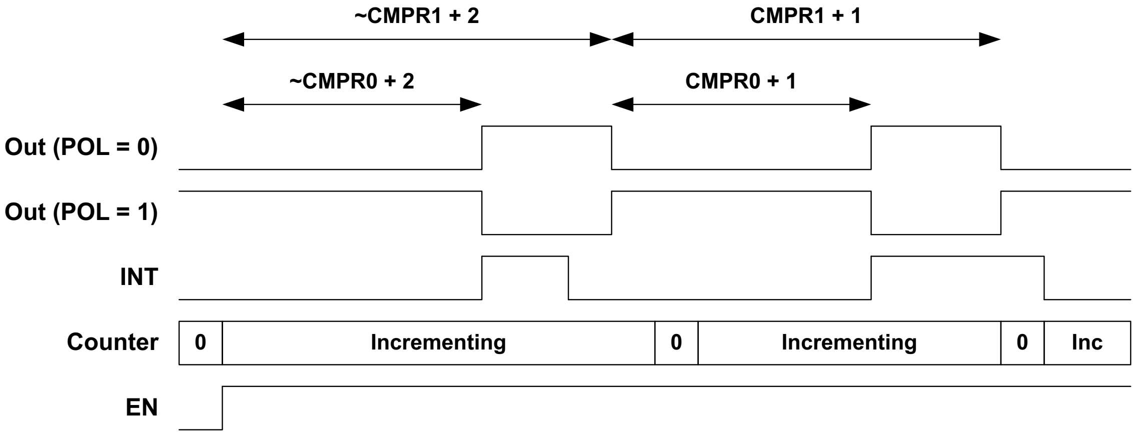


**Figure 43. Counter/Timer Operation, FN = 3**

### 8.2.5 *Continuous (FN = 4)*

Operation in this mode is shown in Figure 44. When the Timer is enabled, the pin output is at the level selected by the POL bit and the Timer is at zero because CLR has been asserted previously. The Timer counts up on each selected clock, and when it reaches the value in the corresponding CMPR0 Register the output pin switches polarity (if the PE bit is set) and an interrupt is generated (if the IE bit is set). The Timer continues to count and is never automatically reset. If the Timer rolls over to zero and reaches the CMPR0 value again, an interrupt will not be generated and the output pin will not change.

This mode is primarily used for two functions. The first is counting transitions on the external input pin, and it may be valuable to generate an interrupt when a specified number of transitions have been detected. The second is as a general timer which software reads in order to measure time periods. In this second case an interrupt is often not used and will not be enabled.

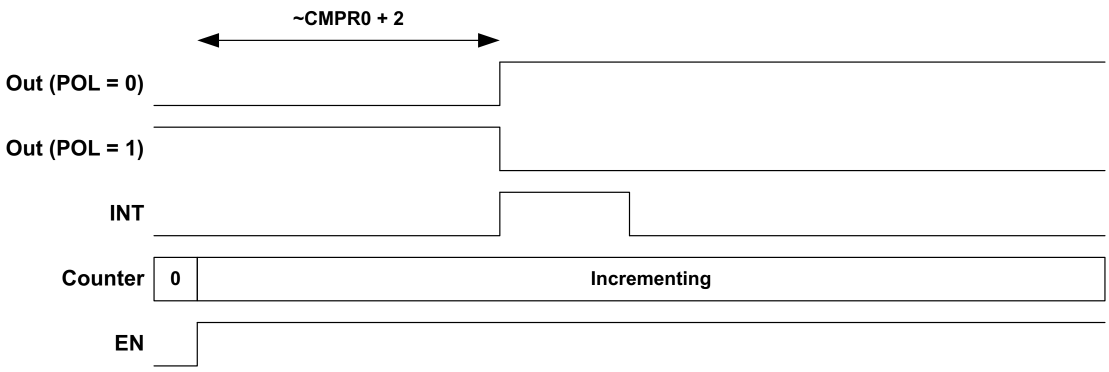


**Figure 44. Counter/Timer Operation, FN = 4**

## 8.3 Creating 32-bit Counters

Each pair (A/B) of 16-bit counters may be combined to create a 32-bit counter. This configuration is created by setting the CTIMER_CTCTRLx_CTRLINKx bit for the pair. The control bits for the A counter of the pair are used to control the 32-bit counter, and the B control bits are ignored. The CMPR0 and CMPR1 registers for each 16-bit counter are concatenated (e.g., CMPR0A3 + (CMPR0B3 << 16)) to provide the 32-bit comparison values, and all timer modes are supported.

## 8.4 Measuring Buck Converter Charge Insertion

Each counter may be connected to a pulse stream from one of the two Buck Converters. One pulse is generated each time the Buck Converter delivers charge onto the capacitor, and therefore the number of pulses is a good indication of the amount of energy used by the corresponding power domain in a particular time period..

Following is a possible option to determine energy consumption. Two counters could be configured with FN = 4 so that they count continuously. One is supplied a Buck Converter pulse stream as its clock, and the other is supplied with a divided version of the LFRC clock to avoid creating extra power consumption due to the power measurement. Once configured such, the two counters should be enabled simultaneously, and after some period of system operation they should be disabled and read. The LFRC count value would now define how much real time has elapsed, and the Buck Converter count value would define how much energy was consumed in that time.

## 8.5 Generating the Sample Rate for the ADC

Timer CTTMRA3 has a special function which allows it to function as the sample trigger generator for the ADC. If the CTIMER_ CTCTRL3 _ADCEN bit is set, the output of the timer is sent to the ADC which uses it as a trigger. Typically, Ctimer3 is configured in Repeated Count(FN =1) mode. TMRA3IE may be set to generate an interrupt whenever the trigger occurs, but typically the ADC interrupt will be used for this purpose.

## 8.6 CTIMER Registers

**Counter/Timer**

**INSTANCE 0 BASE ADDRESS:**0x40008000

The Counter/Timer block contains 8 sixteen bit counter or timer functions. Each pair of these counters can be cascaded into 32 bit Counter/Timer functions.

### *8.6.1 Register Memory Map*

**Table 391: CTIMER Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0x40008000 | TMR0 | Counter/Timer Register |
| 0x40008004 | CMPRA0 | Counter/Timer A0 Compare Registers |
| 0x40008008 | CMPRB0 | Counter/Timer B0 Compare Registers |
| 0x4000800C | CTRL0 | Counter/Timer Control |
| 0x40008010 | TMR1 | Counter/Timer Register |
| 0x40008014 | CMPRA1 | Counter/Timer A1 Compare Registers |
| 0x40008018 | CMPRB1 | Counter/Timer B1 Compare Registers |
| 0x4000801C | CTRL1 | Counter/Timer Control |
| 0x40008020 | TMR2 | Counter/Timer Register |
| 0x40008024 | CMPRA2 | Counter/Timer A2 Compare Registers |
| 0x40008028 | CMPRB2 | Counter/Timer B2 Compare Registers |
| 0x4000802C | CTRL2 | Counter/Timer Control |
| 0x40008030 | TMR3 | Counter/Timer Register |
| 0x40008034 | CMPRA3 | Counter/Timer A3 Compare Registers |
| 0x40008038 | CMPRB3 | Counter/Timer B3 Compare Registers |
| 0x4000803C | CTRL3 | Counter/Timer Control |
| 0x40008200 | INTEN | Counter/Timer Interrupts: Enable |
| 0x40008204 | INTSTAT | Counter/Timer Interrupts: Status |
| 0x40008208 | INTCLR | Counter/Timer Interrupts: Clear |
| 0x4000820C | INTSET | Counter/Timer Interrupts: Set |

### *8.6.2 CTIMER Registers*

#### *8.6.2.1 TMR0 Register*

**Counter/Timer Register**

**OFFSET:** 0x00000000

**INSTANCE 0 ADDRESS:** 0x40008000

This register holds the running time or event count, either for each 16 bit half or for the whole 32 bit count when the pair is linked.

**Table 392: TMR0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CTTMRB0 | | | | | | | | | | | | | | | | CTTMRA0 | | | | | | | | | | | | | | | |

**Table 393: TMR0 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | CTTMRB0 | 0x0 | RO | Counter/Timer B0. |
| 15:0 | CTTMRA0 | 0x0 | RO | Counter/Timer A0. |

#### *8.6.2.2 CMPRA0 Register*

**Counter/Timer A0 Compare Registers**

**OFFSET:** 0x00000004

**INSTANCE 0 ADDRESS:** 0x40008004

Compare limits for timer half A.

**Table 394: CMPRA0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMPR1A0 | | | | | | | | | | | | | | | | CMPR0A0 | | | | | | | | | | | | | | | |

**Table 395: CMPRA0 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | CMPR1A0 | 0x0 | RW | Counter/Timer A0 Compare Register 1. Holds the upper limit for timer half A. |

**Table 395: CMPRA0 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 15:0 | CMPR0A0 | 0x0 | RW | Counter/Timer A0 Compare Register 0. Holds the lower limit for timer half A. |

### 8.6.2.3 CMPRB0 Register

**Counter/Timer B0 Compare Registers**

**OFFSET:** 0x00000008

**INSTANCE 0 ADDRESS:** 0x40008008

Compare limits for timer half B.

**Table 396: CMPRB0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMPR1B0 | | | | | | | | | | | | | | | | CMPR0B0 | | | | | | | | | | | | | | | |

**Table 397: CMPRB0 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | CMPR1B0 | 0x0 | RW | Counter/Timer B0 Compare Register 1. Holds the upper limit for timer half B. |
| 15:0 | CMPR0B0 | 0x0 | RW | Counter/Timer B0 Compare Register 0. Holds the lower limit for timer half B. |

### 8.6.2.4 CTRL0 Register

**Counter/Timer Control**

**OFFSET:** 0x0000000C

**INSTANCE 0 ADDRESS:** 0x4000800C

Control bit fields for both halves of timer 0.

**Table 398: CTRL0 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CTLINK0 | RSVD | | TMRB0POL | TMRB0CLR | TMRB0PE | TMRB0IE | TMRB0FN | | | TMRB0CLK | | | | | TMRB0EN | RSVD | | | TMRA0POL | TMRA0CLR | TMRA0PE | TMRA0IE | TMRA0FN | | | TMRA0CLK | | | | | TMRA0EN |

**Table 399: CTRL0 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | CTLINK0 | 0x0 | RW | Counter/Timer A0/B0 Link bit.<br><br>TWO_16BIT_TIMERS = 0x0 - Use A0/B0 timers as two independent 16-bit timers (default).<br>32BIT_TIMER = 0x1 - Link A0/B0 timers into a single 32-bit timer. |
| 30:29 | RSVD | 0x0 | RO | RESERVED |
| 28 | TMRB0POL | 0x0 | RW | Counter/Timer B0 output polarity.<br><br>NORMAL = 0x0 - The polarity of the TMRPINB0 pin is the same as the timer output.<br>INVERTED = 0x1 - The polarity of the TMRPINB0 pin is the inverse of the timer output. |
| 27 | TMRB0CLR | 0x0 | RW | Counter/Timer B0 Clear bit.<br><br>RUN = 0x0 - Allow counter/timer B0 to run<br>CLEAR = 0x1 - Holds counter/timer B0 at 0x0000. |
| 26 | TMRB0PE | 0x0 | RW | Counter/Timer B0 Output Enable bit.<br><br>DIS = 0x0 - Counter/Timer B holds the TMRPINB signal at the value TMRB0POL.<br>EN = 0x1 - Enable counter/timer B0 to generate a signal on TMRPINB. |
| 25 | TMRB0IE | 0x0 | RW | Counter/Timer B0 Interrupt Enable bit.<br><br>DIS = 0x0 - Disable counter/timer B0 from generating an interrupt.<br>EN = 0x1 - Enable counter/timer B0 to generate an interrupt. |
| 24:22 | TMRB0FN | 0x0 | RW | Counter/Timer B0 Function Select.<br><br>SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B0, stop.<br>REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B0, restart.<br>PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B0, assert, count to CMPR1B, deassert, stop.<br>PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0B0, assert, count to CMPR1B0, deassert, restart.<br>CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously. |

### Table 399: CTRL0 Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 21:17 | TMRB0CLK | 0x0 | RW | Counter/Timer B0 Clock Select.<br><br>TMRPIN = 0x0 - Clock source is TMRPINB.<br>HFRC = 0x1 - Clock source is the HFRC<br>HFRC_DIV8 = 0x2 - Clock source is HFRC / 8<br>HFRC_DIV128 = 0x3 - Clock source is HFRC / 128<br>HFRC_DIV512 = 0x4 - Clock source is HFRC / 512<br>HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048<br>XT = 0x6 - Clock source is the XT (uncalibrated).<br>XT_DIV2 = 0x7 - Clock source is XT / 2<br>XT_DIV16 = 0x8 - Clock source is XT / 16<br>XT_DIV256 = 0x9 - Clock source is XT / 256<br>LFRC_DIV2 = 0xA - Clock source is LFRC / 2<br>LFRC_DIV32 = 0xB - Clock source is LFRC / 32<br>LFRC_DIV1K = 0xC - Clock source is LFRC / 1024<br>LFRC = 0xD - Clock source is LFRC / 16K<br>RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator.<br>HCLK = 0xF - Clock source is HCLK.<br>BUCKB = 0x10 - Clock source is buck converter stream B. |
| 16 | TMRB0EN | 0x0 | RW | Counter/Timer B0 Enable bit.<br><br>DIS = 0x0 - Counter/Timer B0 Disable.<br>EN = 0x1 - Counter/Timer B0 Enable. |
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12 | TMRA0POL | 0x0 | RW | Counter/Timer A0 output polarity.<br><br>NORMAL = 0x0 - The polarity of the TMRPINA0 pin is the same as the timer output.<br>INVERTED = 0x1 - The polarity of the TMRPINA0 pin is the inverse of the timer output. |
| 11 | TMRA0CLR | 0x0 | RW | Counter/Timer A0 Clear bit.<br><br>RUN = 0x0 - Allow counter/timer A0 to run<br>CLEAR = 0x1 - Holds counter/timer A0 at 0x0000. |
| 10 | TMRA0PE | 0x0 | RW | Counter/Timer A0 Output Enable bit.<br><br>DIS = 0x0 - Counter/Timer A holds the TMRPINA signal at the value TMRA0POL.<br>EN = 0x1 - Enable counter/timer B0 to generate a signal on TMRPINB. |
| 9 | TMRA0IE | 0x0 | RW | Counter/Timer A0 Interrupt Enable bit.<br><br>DIS = 0x0 - Disable counter/timer A0 from generating an interrupt.<br>EN = 0x1 - Enable counter/timer A0 to generate an interrupt. |
| 8:6 | TMRA0FN | 0x0 | RW | Counter/Timer A0 Function Select.<br><br>SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A0, stop.<br>REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A0, restart.<br>PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A0, assert, count to CMPR1B, deassert, stop.<br>PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0A0, assert, count to CMPR1A0, deassert, restart.<br>CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously. |

**Table 399: CTRL0 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 5:1 | TMRA0CLK | 0x0 | RW | Counter/Timer A0 Clock Select.<br><br>TMRPIN = 0x0 - Clock source is TMRPINA.<br>HFRC = 0x1 - Clock source is the HFRC<br>HFRC_DIV8 = 0x2 - Clock source is HFRC / 8<br>HFRC_DIV128 = 0x3 - Clock source is HFRC / 128<br>HFRC_DIV512 = 0x4 - Clock source is HFRC / 512<br>HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048<br>XT = 0x6 - Clock source is the XT (uncalibrated).<br>XT_DIV2 = 0x7 - Clock source is XT / 2<br>XT_DIV16 = 0x8 - Clock source is XT / 16<br>XT_DIV256 = 0x9 - Clock source is XT / 256<br>LFRC_DIV2 = 0xA - Clock source is LFRC / 2<br>LFRC_DIV32 = 0xB - Clock source is LFRC / 32<br>LFRC_DIV1K = 0xC - Clock source is LFRC / 1024<br>LFRC = 0xD - Clock source is LFRC / 16K<br>RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator.<br>HCLK = 0xF - Clock source is HCLK.<br>BUCKA = 0x10 - Clock source is buck converter stream A. |
| 0 | TMRA0EN | 0x0 | RW | Counter/Timer A0 Enable bit.<br><br>DIS = 0x0 - Counter/Timer A0 Disable.<br>EN = 0x1 - Counter/Timer A0 Enable. |

### *8.6.2.5 TMR1 Register*

**Counter/Timer Register**

**OFFSET:** 0x00000010

**INSTANCE 0 ADDRESS:** 0x40008010

This register holds the running time or event count, either for each 16 bit half or for the whole 32 bit count when the pair is linked.

**Table 400: TMR1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CTTMRB1 | | | | | | | | | | | | | | | | CTTMRA1 | | | | | | | | | | | | | | | |

**Table 401: TMR1 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | CTTMRB1 | 0x0 | RO | Counter/Timer B1. |
| 15:0 | CTTMRA1 | 0x0 | RO | Counter/Timer A1. |

#### *8.6.2.6 CMPRA1 Register*

**Counter/Timer A1 Compare Registers**

**OFFSET:** 0x00000014

**INSTANCE 0 ADDRESS:** 0x40008014

This register holds the compare limits for timer half A.

**Table 402: CMPRA1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMPR1A1 | | | | | | | | | | | | | | | | CMPR0A1 | | | | | | | | | | | | | | | |

**Table 403: CMPRA1 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | CMPR1A1 | 0x0 | RW | Counter/Timer A1 Compare Register 1. |
| 15:0 | CMPR0A1 | 0x0 | RW | Counter/Timer A1 Compare Register 0. |

#### *8.6.2.7 CMPRB1 Register*

**Counter/Timer B1 Compare Registers**

**OFFSET:** 0x00000018

**INSTANCE 0 ADDRESS:** 0x40008018

This register holds the compare limits for timer half B.

**Table 404: CMPRB1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMPR1B1 | | | | | | | | | | | | | | | | CMPR0B1 | | | | | | | | | | | | | | | |

**Table 405: CMPRB1 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | CMPR1B1 | 0x0 | RW | Counter/Timer B1 Compare Register 1. |
| 15:0 | CMPR0B1 | 0x0 | RW | Counter/Timer B1 Compare Register 0. |

### *8.6.2.8 CTRL1 Register*

**Counter/Timer Control**

**OFFSET:** 0x0000001C

**INSTANCE 0 ADDRESS:** 0x4000801C

This register holds the control bit fields for both halves of timer 1.

**Table 406: CTRL1 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CTLINK1 | RSVD | | TMRB1POL | TMRB1CLR | TMRB1PE | TMRB1IE | TMRB1FN | | | TMRB1CLK | | | | | TMRB1EN | RSVD | | | TMRA1POL | TMRA1CLR | TMRA1PE | TMRA1IE | TMRA1FN | | | TMRA1CLK | | | | | TMRA1EN |

**Table 407: CTRL1 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | CTLINK1 | 0x0 | RW | Counter/Timer A1/B1 Link bit.<br><br>TWO_16BIT_TIMERS = 0x0 - Use A0/B0 timers as two independent 16-bit timers (default).<br>32BIT_TIMER = 0x1 - Link A1/B1 timers into a single 32-bit timer. |
| 30:29 | RSVD | 0x0 | RO | RESERVED |
| 28 | TMRB1POL | 0x0 | RW | Counter/Timer B1 output polarity.<br><br>NORMAL = 0x0 - The polarity of the TMRPINB1 pin is the same as the timer output.<br>INVERTED = 0x1 - The polarity of the TMRPINB1 pin is the inverse of the timer output. |
| 27 | TMRB1CLR | 0x0 | RW | Counter/Timer B1 Clear bit.<br><br>RUN = 0x0 - Allow counter/timer B1 to run<br>CLEAR = 0x1 - Holds counter/timer B1 at 0x0000. |
| 26 | TMRB1PE | 0x0 | RW | Counter/Timer B1 Output Enable bit.<br><br>DIS = 0x0 - Counter/Timer B holds the TMRPINB signal at the value TMRB1POL.<br>EN = 0x1 - Enable counter/timer B1 to generate a signal on TMRPINB. |
| 25 | TMRB1IE | 0x0 | RW | Counter/Timer B1 Interrupt Enable bit.<br><br>DIS = 0x0 - Disable counter/timer B1 from generating an interrupt.<br>EN = 0x1 - Enable counter/timer B1 to generate an interrupt. |

**Table 407: CTRL1 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 24:22 | TMRB1FN | 0x0 | RW | Counter/Timer B1 Function Select.<br><br>SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B1, stop.<br>REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B1, restart.<br>PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B1, assert, count to CMPR1B, deassert, stop.<br>PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0B1, assert, count to CMPR1B1, deassert, restart.<br>CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously. |
| 21:17 | TMRB1CLK | 0x0 | RW | Counter/Timer B1 Clock Select.<br><br>TMRPIN = 0x0 - Clock source is TMRPINB.<br>HFRC = 0x1 - Clock source is the HFRC<br>HFRC_DIV8 = 0x2 - Clock source is HFRC / 8<br>HFRC_DIV128 = 0x3 - Clock source is HFRC / 128<br>HFRC_DIV512 = 0x4 - Clock source is HFRC / 512<br>HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048<br>XT = 0x6 - Clock source is the XT (uncalibrated).<br>XT_DIV2 = 0x7 - Clock source is XT / 2<br>XT_DIV16 = 0x8 - Clock source is XT / 16<br>XT_DIV256 = 0x9 - Clock source is XT / 256<br>LFRC_DIV2 = 0xA - Clock source is LFRC / 2<br>LFRC_DIV32 = 0xB - Clock source is LFRC / 32<br>LFRC_DIV1K = 0xC - Clock source is LFRC / 1024<br>LFRC = 0xD - Clock source is LFRC / 16K<br>RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator.<br>HCLK = 0xF - Clock source is HCLK.<br>BUCKB = 0x10 - Clock source is buck converter stream B. |
| 16 | TMRB1EN | 0x0 | RW | Counter/Timer B1 Enable bit.<br><br>DIS = 0x0 - Counter/Timer B1 Disable.<br>EN = 0x1 - Counter/Timer B1 Enable. |
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12 | TMRA1POL | 0x0 | RW | Counter/Timer A1 output polarity.<br><br>NORMAL = 0x0 - The polarity of the TMRPINA1 pin is the same as the timer output.<br>INVERTED = 0x1 - The polarity of the TMRPINA1 pin is the inverse of the timer output. |
| 11 | TMRA1CLR | 0x0 | RW | Counter/Timer A1 Clear bit.<br><br>RUN = 0x0 - Allow counter/timer A1 to run<br>CLEAR = 0x1 - Holds counter/timer A1 at 0x0000. |
| 10 | TMRA1PE | 0x0 | RW | Counter/Timer A1 Output Enable bit.<br><br>DIS = 0x0 - Counter/Timer A holds the TMRPINA signal at the value TMRA1POL.<br>EN = 0x1 - Enable counter/timer A1 to generate a signal on TMRPINA. |
| 9 | TMRA1IE | 0x0 | RW | Counter/Timer A1 Interrupt Enable bit.<br><br>DIS = 0x0 - Disable counter/timer A1 from generating an interrupt.<br>EN = 0x1 - Enable counter/timer A1 to generate an interrupt. |

**Table 407: CTRL1 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 8:6 | TMRA1FN | 0x0 | RW | Counter/Timer A1 Function Select.<br><br>SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A1, stop.<br>REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A1, restart.<br>PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A1, assert, count to CMPR1B, deassert, stop.<br>PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0A1, assert, count to CMPR1A1, deassert, restart.<br>CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously. |
| 5:1 | TMRA1CLK | 0x0 | RW | Counter/Timer A1 Clock Select.<br><br>TMRPIN = 0x0 - Clock source is TMRPINA.<br>HFRC = 0x1 - Clock source is the HFRC<br>HFRC_DIV8 = 0x2 - Clock source is the HFRC / 8<br>HFRC_DIV128 = 0x3 - Clock source is HFRC / 128<br>HFRC_DIV512 = 0x4 - Clock source is HFRC / 512<br>HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048<br>XT = 0x6 - Clock source is the XT (uncalibrated).<br>XT_DIV2 = 0x7 - Clock source is XT / 2<br>XT_DIV16 = 0x8 - Clock source is XT / 16<br>XT_DIV256 = 0x9 - Clock source is XT / 256<br>LFRC_DIV2 = 0xA - Clock source is LFRC / 2<br>LFRC_DIV32 = 0xB - Clock source is LFRC / 32<br>LFRC_DIV1K = 0xC - Clock source is LFRC / 1024<br>LFRC = 0xD - Clock source is LFRC / 16K<br>RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator.<br>HCLK = 0xF - Clock source is HCLK.<br>BUCKA = 0x10 - Clock source is buck converter stream A. |
| 0 | TMRA1EN | 0x0 | RW | Counter/Timer A1 Enable bit.<br><br>DIS = 0x0 - Counter/Timer A1 Disable.<br>EN = 0x1 - Counter/Timer A1 Enable. |

### 8.6.2.9 *TMR2 Register*

**Counter/Timer Register**

**OFFSET:** 0x00000020

**INSTANCE 0 ADDRESS:** 0x40008020

Counter/Timer Register

**Table 408: TMR2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CTTMRB2 | | | | | | | | | | | | | | | | CTTMRA2 | | | | | | | | | | | | | | | |

**Table 409: TMR2 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | CTTMRB2 | 0x0 | RO | Counter/Timer B2. |
| 15:0 | CTTMRA2 | 0x0 | RO | Counter/Timer A2. |

### *8.6.2.10 CMPRA2 Register*

**Counter/Timer A2 Compare Registers**

**OFFSET:** 0x00000024

**INSTANCE 0 ADDRESS:** 0x40008024

This register holds the compare limits for timer half A.

**Table 410: CMPRA2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMPR1A2 | | | | | | | | | | | | | | | | CMPR0A2 | | | | | | | | | | | | | | | |

**Table 411: CMPRA2 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | CMPR1A2 | 0x0 | RW | Counter/Timer A2 Compare Register 1. |
| 15:0 | CMPR0A2 | 0x0 | RW | Counter/Timer A2 Compare Register 0. |

### *8.6.2.11 CMPRB2 Register*

**Counter/Timer B2 Compare Registers**

**OFFSET:** 0x00000028

**INSTANCE 0 ADDRESS:** 0x40008028

This register holds the compare limits for timer half B.

**Table 412: CMPRB2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMPR1B2 | | | | | | | | | | | | | | | | CMPR0B2 | | | | | | | | | | | | | | | |

**Table 413: CMPRB2 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | CMPR1B2 | 0x0 | RW | Counter/Timer B2 Compare Register 1. |
| 15:0 | CMPR0B2 | 0x0 | RW | Counter/Timer B2 Compare Register 0. |

### *8.6.2.12 CTRL2 Register*

**Counter/Timer Control**

**OFFSET:** 0x0000002C

**INSTANCE 0 ADDRESS:** 0x4000802C

This register holds the control bit fields for both halves of timer 2.

**Table 414: CTRL2 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CTLINK2 | RSVD | | TMRB2POL | TMRB2CLR | TMRB2PE | TMRB2IE | TMRB2FN | | | TMRB2CLK | | | | | TMRB2EN | RSVD | | | TMRA2POL | TMRA2CLR | TMRA2PE | TMRA2IE | TMRA2FN | | | TMRA2CLK | | | | | TMRA2EN |

**Table 415: CTRL2 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | CTLINK2 | 0x0 | RW | Counter/Timer A2/B2 Link bit.<br><br>TWO_16BIT_TIMERS = 0x0 - Use A0/B0 timers as two independent 16-bit timers (default).<br>32BIT_TIMER = 0x1 - Link A2/B2 timers into a single 32-bit timer. |
| 30:29 | RSVD | 0x0 | RO | RESERVED |
| 28 | TMRB2POL | 0x0 | RW | Counter/Timer B2 output polarity.<br><br>NORMAL = 0x0 - The polarity of the TMRPINB2 pin is the same as the timer output.<br>INVERTED = 0x1 - The polarity of the TMRPINB2 pin is the inverse of the timer output. |
| 27 | TMRB2CLR | 0x0 | RW | Counter/Timer B2 Clear bit.<br><br>RUN = 0x0 - Allow counter/timer B2 to run<br>CLEAR = 0x1 - Holds counter/timer B2 at 0x0000. |

### Table 415: CTRL2 Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 26 | TMRB2PE | 0x0 | RW | Counter/Timer B2 Output Enable bit.<br><br>DIS = 0x0 - Counter/Timer B holds the TMRPINB signal at the value TMRB2POL.<br>EN = 0x1 - Enable counter/timer B2 to generate a signal on TMRPINB. |
| 25 | TMRB2IE | 0x0 | RW | Counter/Timer B2 Interrupt Enable bit.<br><br>DIS = 0x0 - Disable counter/timer B2 from generating an interrupt.<br>EN = 0x1 - Enable counter/timer B2 to generate an interrupt. |
| 24:22 | TMRB2FN | 0x0 | RW | Counter/Timer B2 Function Select.<br><br>SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B2, stop.<br>REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B2, restart.<br>PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B2, assert, count to CMPR1B, deassert, stop.<br>PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0B2, assert, count to CMPR1B2, deassert, restart.<br>CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously. |
| 21:17 | TMRB2CLK | 0x0 | RW | Counter/Timer B2 Clock Select.<br><br>TMRPIN = 0x0 - Clock source is TMRPINB.<br>HFRC = 0x1 - Clock source is the HFRC<br>HFRC_DIV8 = 0x2 - Clock source is HFRC / 8<br>HFRC_DIV128 = 0x3 - Clock source is HFRC / 128<br>HFRC_DIV512 = 0x4 - Clock source is HFRC / 512<br>HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048<br>XT = 0x6 - Clock source is the XT (uncalibrated).<br>XT_DIV2 = 0x7 - Clock source is XT / 2<br>XT_DIV16 = 0x8 - Clock source is XT / 16<br>XT_DIV256 = 0x9 - Clock source is XT / 256<br>LFRC_DIV2 = 0xA - Clock source is LFRC / 2<br>LFRC_DIV32 = 0xB - Clock source is LFRC / 32<br>LFRC_DIV1K = 0xC - Clock source is LFRC / 1024<br>LFRC = 0xD - Clock source is LFRC / 16K<br>RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator.<br>HCLK = 0xF - Clock source is HCLK.<br>BUCKA = 0x10 - Clock source is buck converter stream A. |
| 16 | TMRB2EN | 0x0 | RW | Counter/Timer B2 Enable bit.<br><br>DIS = 0x0 - Counter/Timer B2 Disable.<br>EN = 0x1 - Counter/Timer B2 Enable. |
| 15:13 | RSVD | 0x0 | RO | RESERVED |
| 12 | TMRA2POL | 0x0 | RW | Counter/Timer A2 output polarity.<br><br>NORMAL = 0x0 - The polarity of the TMRPINA2 pin is the same as the timer output.<br>INVERTED = 0x1 - The polarity of the TMRPINA2 pin is the inverse of the timer output. |
| 11 | TMRA2CLR | 0x0 | RW | Counter/Timer A2 Clear bit.<br><br>RUN = 0x0 - Allow counter/timer A2 to run<br>CLEAR = 0x1 - Holds counter/timer A2 at 0x0000. |

### Table 415: CTRL2 Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 10 | TMRA2PE | 0x0 | RW | Counter/Timer A2 Output Enable bit.<br><br>DIS = 0x0 - Counter/Timer A holds the TMRPINA signal at the value TMRA2POL.<br>EN = 0x1 - Enable counter/timer A2 to generate a signal on TMRPINA. |
| 9 | TMRA2IE | 0x0 | RW | Counter/Timer A2 Interrupt Enable bit.<br><br>DIS = 0x0 - Disable counter/timer A2 from generating an interrupt.<br>EN = 0x1 - Enable counter/timer A2 to generate an interrupt. |
| 8:6 | TMRA2FN | 0x0 | RW | Counter/Timer A2 Function Select.<br><br>SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A2, stop.<br>REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A2, restart.<br>PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A2, assert, count to CMPR1B, deassert, stop.<br>PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0A2, assert, count to CMPR1A2, deassert, restart.<br>CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously. |
| 5:1 | TMRA2CLK | 0x0 | RW | Counter/Timer A2 Clock Select.<br><br>TMRPIN = 0x0 - Clock source is TMRPINA.<br>HFRC = 0x1 - Clock source is the HFRC<br>HFRC_DIV8 = 0x2 - Clock source is HFRC / 8<br>HFRC_DIV128 = 0x3 - Clock source is HFRC / 128<br>HFRC_DIV512 = 0x4 - Clock source is HFRC / 512<br>HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048<br>XT = 0x6 - Clock source is the XT (uncalibrated).<br>XT_DIV2 = 0x7 - Clock source is XT / 2<br>XT_DIV16 = 0x8 - Clock source is XT / 16<br>XT_DIV256 = 0x9 - Clock source is XT / 256<br>LFRC_DIV2 = 0xA - Clock source is LFRC / 2<br>LFRC_DIV32 = 0xB - Clock source is LFRC / 32<br>LFRC_DIV1K = 0xC - Clock source is LFRC / 1024<br>LFRC = 0xD - Clock source is LFRC / 16K<br>RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator.<br>HCLK = 0xF - Clock source is HCLK.<br>BUCKB = 0x10 - Clock source is buck converter stream B. |
| 0 | TMRA2EN | 0x0 | RW | Counter/Timer A2 Enable bit.<br><br>DIS = 0x0 - Counter/Timer A2 Disable.<br>EN = 0x1 - Counter/Timer A2 Enable. |

### *8.6.2.13 TMR3 Register*

**Counter/Timer Register**

**OFFSET:** 0x00000030

**INSTANCE 0 ADDRESS:** 0x40008030

Counter/Timer Register

**Table 416: TMR3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CTTMRB3 | | | | | | | | | | | | | | | | CTTMRA3 | | | | | | | | | | | | | | | |

**Table 417: TMR3 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | CTTMRB3 | 0x0 | RO | Counter/Timer B3. |
| 15:0 | CTTMRA3 | 0x0 | RO | Counter/Timer A3. |

### *8.6.2.14 CMPRA3 Register*

**Counter/Timer A3 Compare Registers**

**OFFSET:** 0x00000034

**INSTANCE 0 ADDRESS:** 0x40008034

This register holds the compare limits for timer half A.

**Table 418: CMPRA3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMPR1A3 | | | | | | | | | | | | | | | | CMPR0A3 | | | | | | | | | | | | | | | |

**Table 419: CMPRA3 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | CMPR1A3 | 0x0 | RW | Counter/Timer A3 Compare Register 1. |
| 15:0 | CMPR0A3 | 0x0 | RW | Counter/Timer A3 Compare Register 0. |

### *8.6.2.15 CMPRB3 Register*

**Counter/Timer B3 Compare Registers**

**OFFSET:** 0x00000038

**INSTANCE 0 ADDRESS:** 0x40008038

This register holds the compare limits for timer half B.

**Table 420: CMPRB3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CMPR1B3 | | | | | | | | | | | | | | | | CMPR0B3 | | | | | | | | | | | | | | | |

**Table 421: CMPRB3 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | CMPR1B3 | 0x0 | RW | Counter/Timer B3 Compare Register 1. |
| 15:0 | CMPR0B3 | 0x0 | RW | Counter/Timer B3 Compare Register 0. |

### *8.6.2.16 CTRL3 Register*

**Counter/Timer Control**

**OFFSET:** 0x0000003C

**INSTANCE 0 ADDRESS:** 0x4000803C

This register holds the control bit fields for both halves of timer 3.

**Table 422: CTRL3 Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CTLINK3 | RSVD | | TMRB3POL | TMRB3CLR | TMRB3PE | TMRB3IE | TMRB3FN | | | TMRB3CLK | | | | | TMRB3EN | ADCEN | RSVD | | TMRA3POL | TMRA3CLR | TMRA3PE | TMRA3IE | TMRA3FN | | | TMRA3CLK | | | | | TMRA3EN |

**Table 423: CTRL3 Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31 | CTLINK3 | 0x0 | RW | Counter/Timer A/B Link bit.<br><br>TWO_16BIT_TIMERS = 0x0 - Use A0/B0 timers as two independent 16-bit timers (default).<br>32BIT_TIMER = 0x1 - Link A3/B3 timers into a single 32-bit timer. |
| 30:29 | RSVD | 0x0 | RO | RESERVED |
| 28 | TMRB3POL | 0x0 | RW | Counter/Timer B3 output polarity.<br><br>NORMAL = 0x0 - The polarity of the TMRPINB3 pin is the same as the timer output.<br>INVERTED = 0x1 - The polarity of the TMRPINB3 pin is the inverse of the timer output. |
| 27 | TMRB3CLR | 0x0 | RW | Counter/Timer B3 Clear bit.<br><br>RUN = 0x0 - Allow counter/timer B3 to run.<br>CLEAR = 0x1 - Holds counter/timer B3 at 0x0000. |
| 26 | TMRB3PE | 0x0 | RW | Counter/Timer B3 Output Enable bit.<br><br>DIS = 0x0 - Counter/Timer B holds the TMRPINB signal at the value TMRB3POL.<br>EN = 0x1 - Enable counter/timer B3 to generate a signal on TMRPINB. |
| 25 | TMRB3IE | 0x0 | RW | Counter/Timer B3 Interrupt Enable bit.<br><br>DIS = 0x0 - Disable counter/timer B3 from generating an interrupt.<br>EN = 0x1 - Enable counter/timer B3 to generate an interrupt. |
| 24:22 | TMRB3FN | 0x0 | RW | Counter/Timer B3 Function Select.<br><br>SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B3, stop.<br>REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B3, restart.<br>PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B3, assert, count to CMPR1B, deassert, stop.<br>PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0B3, assert, count to CMPR1B3, deassert, restart.<br>CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously. |

### Table 423: CTRL3 Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 21:17 | TMRB3CLK | 0x0 | RW | Counter/Timer B3 Clock Select.<br><br>TMRPIN = 0x0 - Clock source is TMRPINB.<br>HFRC = 0x1 - Clock source is the HFRC<br>HFRC_DIV8 = 0x2 - Clock source is HFRC / 8<br>HFRC_DIV128 = 0x3 - Clock source is HFRC / 128<br>HFRC_DIV512 = 0x4 - Clock source is HFRC / 512<br>HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048<br>XT = 0x6 - Clock source is the XT (uncalibrated).<br>XT_DIV2 = 0x7 - Clock source is XT / 2<br>XT_DIV16 = 0x8 - Clock source is XT / 16<br>XT_DIV256 = 0x9 - Clock source is XT / 256<br>LFRC_DIV2 = 0xA - Clock source is LFRC / 2<br>LFRC_DIV32 = 0xB - Clock source is LFRC / 32<br>LFRC_DIV1K = 0xC - Clock source is LFRC / 1024<br>LFRC = 0xD - Clock source is LFRC / 16K<br>RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator.<br>HCLK = 0xF - Clock source is HCLK.<br>BUCKA = 0x10 - Clock source is buck converter stream A. |
| 16 | TMRB3EN | 0x0 | RW | Counter/Timer B3 Enable bit.<br><br>DIS = 0x0 - Counter/Timer B3 Disable.<br>EN = 0x1 - Counter/Timer B3 Enable. |
| 15 | ADCEN | 0x0 | RW | Special Timer A3 enable for ADC function. |
| 14:13 | RSVD | 0x0 | RO | RESERVED |
| 12 | TMRA3POL | 0x0 | RW | Counter/Timer A3 output polarity.<br><br>NORMAL = 0x0 - The polarity of the TMRPINA3 pin is the same as the timer output.<br>INVERTED = 0x1 - The polarity of the TMRPINA3 pin is the inverse of the timer output. |
| 11 | TMRA3CLR | 0x0 | RW | Counter/Timer A3 Clear bit.<br><br>CLEAR = 0x1 - Holds counter/timer A3 at 0x0000. |
| 10 | TMRA3PE | 0x0 | RW | Counter/Timer A3 Output Enable bit.<br><br>DIS = 0x0 - Counter/Timer A holds the TMRPINA signal at the value TMRA3POL.<br>EN = 0x1 - Enable counter/timer A3 to generate a signal on TMRPINA. |
| 9 | TMRA3IE | 0x0 | RW | Counter/Timer A3 Interrupt Enable bit.<br><br>DIS = 0x0 - Disable counter/timer A3 from generating an interrupt.<br>EN = 0x1 - Enable counter/timer A3 to generate an interrupt. |

Table 423: CTRL3 Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 8:6 | TMRA3FN | 0x0 | RW | Counter/Timer A3 Function Select.<br><br>SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A3, stop.<br>REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A3, restart.<br>PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A3, assert, count to CMPR1B, deassert, stop.<br>PULSE_CONT = 0x3 - Pulse continously. Count to CMPR0A3, assert, count to CMPR1A3, deassert, restart.<br>CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously. |
| 5:1 | TMRA3CLK | 0x0 | RW | Counter/Timer A3 Clock Select.<br><br>TMRPIN = 0x0 - Clock source is TMRPINA.<br>HFRC = 0x1 - Clock source is the HFRC<br>HFRC_DIV8 = 0x2 - Clock source is HFRC / 8<br>HFRC_DIV128 = 0x3 - Clock source is HFRC / 128<br>HFRC_DIV512 = 0x4 - Clock source is HFRC / 512<br>HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048<br>XT = 0x6 - Clock source is the XT (uncalibrated).<br>XT_DIV2 = 0x7 - Clock source is XT / 2<br>XT_DIV16 = 0x8 - Clock source is XT / 16<br>XT_DIV256 = 0x9 - Clock source is XT / 256<br>LFRC_DIV2 = 0xA - Clock source is LFRC / 2<br>LFRC_DIV32 = 0xB - Clock source is LFRC / 32<br>LFRC_DIV1K = 0xC - Clock source is LFRC / 1024<br>LFRC = 0xD - Clock source is LFRC / 16K<br>RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator.<br>HCLK = 0xF - Clock source is HCLK.<br>BUCKB = 0x10 - Clock source is buck converter stream B. |
| 0 | TMRA3EN | 0x0 | RW | Counter/Timer A3 Enable bit.<br><br>DIS = 0x0 - Counter/Timer A3 Disable.<br>EN = 0x1 - Counter/Timer A3 Enable. |

### 8.6.2.17 *INTEN Register*

**Counter/Timer Interrupts: Enable**

**OFFSET:** 0x00000200

**INSTANCE 0 ADDRESS:** 0x40008200

Set bits in this register to allow this module to generate the corresponding interrupt.

Table 424: INTEN Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | CTMRB3INT | CTMRA3INT | CTMRB2INT | CTMRA2INT | CTMRB1INT | CTMRA1INT | CTMRB0INT | CTMRA0INT |

Table 425: INTEN Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED |
| 7 | CTMRB3INT | 0x0 | RW | Counter/Timer B3 interrupt. |
| 6 | CTMRA3INT | 0x0 | RW | Counter/Timer A3 interrupt. |
| 5 | CTMRB2INT | 0x0 | RW | Counter/Timer B2 interrupt. |
| 4 | CTMRA2INT | 0x0 | RW | Counter/Timer A2 interrupt. |
| 3 | CTMRB1INT | 0x0 | RW | Counter/Timer B1 interrupt. |
| 2 | CTMRA1INT | 0x0 | RW | Counter/Timer A1 interrupt. |
| 1 | CTMRB0INT | 0x0 | RW | Counter/Timer B0 interrupt. |
| 0 | CTMRA0INT | 0x0 | RW | Counter/Timer A0 interrupt. |

### *8.6.2.18 INTSTAT Register*

**Counter/Timer Interrupts: Status**

**OFFSET:** 0x00000204

**INSTANCE 0 ADDRESS:** 0x40008204

Read bits from this register to discover the cause of a recent interrupt.

Table 426: INTSTAT Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | CTMRB3INT | CTMRA3INT | CTMRB2INT | CTMRA2INT | CTMRB1INT | CTMRA1INT | CTMRB0INT | CTMRA0INT |

Table 427: INTSTAT Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED |
| 7 | CTMRB3INT | 0x0 | RW | Counter/Timer B3 interrupt. |

**Table 427: INTSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 6 | CTMRA3INT | 0x0 | RW | Counter/Timer A3 interrupt. |
| 5 | CTMRB2INT | 0x0 | RW | Counter/Timer B2 interrupt. |
| 4 | CTMRA2INT | 0x0 | RW | Counter/Timer A2 interrupt. |
| 3 | CTMRB1INT | 0x0 | RW | Counter/Timer B1 interrupt. |
| 2 | CTMRA1INT | 0x0 | RW | Counter/Timer A1 interrupt. |
| 1 | CTMRB0INT | 0x0 | RW | Counter/Timer B0 interrupt. |
| 0 | CTMRA0INT | 0x0 | RW | Counter/Timer A0 interrupt. |

### *8.6.2.19 INTCLR Register*

**Counter/Timer Interrupts: Clear**

**OFFSET:** 0x00000208

**INSTANCE 0 ADDRESS:** 0x40008208

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

**Table 428: INTCLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | CTMRB3INT | CTMRA3INT | CTMRB2INT | CTMRA2INT | CTMRB1INT | CTMRA1INT | CTMRB0INT | CTMRA0INT |

**Table 429: INTCLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED |
| 7 | CTMRB3INT | 0x0 | RW | Counter/Timer B3 interrupt. |
| 6 | CTMRA3INT | 0x0 | RW | Counter/Timer A3 interrupt. |
| 5 | CTMRB2INT | 0x0 | RW | Counter/Timer B2 interrupt. |

Table 429: INTCLR Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 4 | CTMRA2INT | 0x0 | RW | Counter/Timer A2 interrupt. |
| 3 | CTMRB1INT | 0x0 | RW | Counter/Timer B1 interrupt. |
| 2 | CTMRA1INT | 0x0 | RW | Counter/Timer A1 interrupt. |
| 1 | CTMRB0INT | 0x0 | RW | Counter/Timer B0 interrupt. |
| 0 | CTMRA0INT | 0x0 | RW | Counter/Timer A0 interrupt. |

### 8.6.2.20 INTSET Register

**Counter/Timer Interrupts: Set**

**OFFSET:** 0x0000020C

**INSTANCE 0 ADDRESS:** 0x4000820C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

Table 430: INTSET Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | CTMRB3INT | CTMRA3INT | CTMRB2INT | CTMRA2INT | CTMRB1INT | CTMRA1INT | CTMRB0INT | CTMRA0INT |

Table 431: INTSET Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED |
| 7 | CTMRB3INT | 0x0 | RW | Counter/Timer B3 interrupt. |
| 6 | CTMRA3INT | 0x0 | RW | Counter/Timer A3 interrupt. |
| 5 | CTMRB2INT | 0x0 | RW | Counter/Timer B2 interrupt. |
| 4 | CTMRA2INT | 0x0 | RW | Counter/Timer A2 interrupt. |
| 3 | CTMRB1INT | 0x0 | RW | Counter/Timer B1 interrupt. |

**Table 431: INTSET Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 2 | CTMRA1INT | 0x0 | RW | Counter/Timer A1 interrupt. |
| 1 | CTMRB0INT | 0x0 | RW | Counter/Timer B0 interrupt. |
| 0 | CTMRA0INT | 0x0 | RW | Counter/Timer A0 interrupt. |

# 9. Watchdog Timer Module

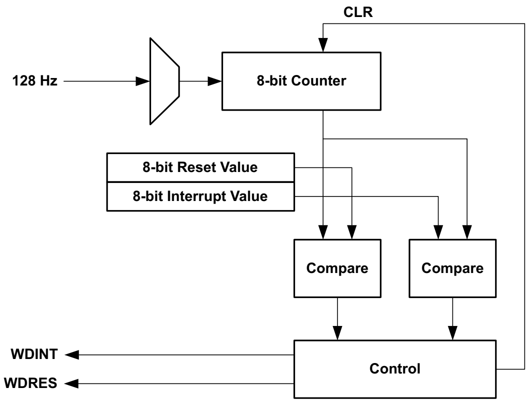


**Figure 45. Block Diagram for the Watchdog Timer Module**

## 9.1 Functional Overview

The Watchdog Timer (WDT), shown in Figure 45, is used to ensure that software is operational, by resetting the Apollo SoC if the WDT reaches a configurable value before being cleared by software. The WDT can be clocked by a 128 Hz clock taken from the low-power LFRC which is always active. The WDT may be locked to ensure that software cannot disable its functionality, in which case the WDTCFG register cannot be updated. An interrupt can also be generated at a different counter value to implement an early warning function.

NOTE

The RESEN bit in the WDTCFG register must be set and the WDREN bit in the RSTCFG register must be set to enable a watchdog timer reset condition.

## 9.2 WDT Registers

**Watchdog Timer**

**INSTANCE 0 BASE ADDRESS:**0x40024000

### 9.2.1 *Register Memory Map*

**Table 432: WDT Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0x40024000 | CFG | Configuration Register |
| 0x40024004 | RSTRT | Restart the watchdog timer |
| 0x40024008 | LOCK | Locks the WDT |
| 0x40024200 | INTEN | WDT Interrupt register: Enable |
| 0x40024204 | INTSTAT | WDT Interrupt register: Status |
| 0x40024208 | INTCLR | WDT Interrupt register: Clear |
| 0x4002420C | INTSET | WDT Interrupt register: Set |

### 9.2.2 *WDT Registers*

#### 9.2.2.1 *CFG Register*

**Configuration Register**

**OFFSET:** 0x00000000

**INSTANCE 0 ADDRESS:** 0x40024000

Configuration Register

**Table 433: CFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | INTVAL | | | | | | | | RESVAL | | | | | | | | RSVD | | | | | RESEN | INTEN | WDTEN |

**Table 434: CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:24 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 23:16 | INTVAL | 0x0 | RW | This bitfield is the compare value for counter bits 7:0 to generate a watchdog interrupt. |
| 15:8 | RESVAL | 0x0 | RW | This bitfield is the compare value for counter bits 7:0 to generate a watchdog reset. |

**Table 434: CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 7:3 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 2 | RESEN | 0x0 | RW | This bitfield enables the WDT reset. |
| 1 | INTEN | 0x0 | RW | This bitfield enables the WDT interrupt. Note : This bit must be set before the interrupt status bit will reflect a watchdog timer expiration. The IER interrupt register must also be enabled for a WDT interrupt to be sent to the NVIC. |
| 0 | WDTEN | 0x0 | RW | This bitfield enables the WDT. |

### *9.2.2.2 RSTRT Register*

**Restart the watchdog timer**

**OFFSET:** 0x00000004

**INSTANCE 0 ADDRESS:** 0x40024004

Restart the watchdog timer

**Table 435: RSTRT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | RSTRT | | | | | | | |

**Table 436: RSTRT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 7:0 | RSTRT | 0x0 | WO | Writing 0xB2 to WDTRSTRT restarts the watchdog timer.<br>KEYVALUE = 0xB2 - This is the key value to write to WDTRSTRT to restart the WDT. |

### 9.2.2.3 *LOCK Register*

**Locks the WDT**

**OFFSET:** 0x00000008

**INSTANCE 0 ADDRESS:** 0x40024008

Locks the WDT

**Table 437: LOCK Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | LOCK | | | | | | | |

**Table 438: LOCK Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 7:0 | LOCK | 0x0 | WO | Writing 0x3A locks the watchdog timer. Once locked, the WDTCFG reg cannot be written and WDTEN is set.<br>KEYVALUE = 0x3A - This is the key value to write to WDTLOCK to lock the WDT. |

### 9.2.2.4 *INTEN Register*

**WDT Interrupt register: Enable**

**OFFSET:** 0x00000200

**INSTANCE 0 ADDRESS:** 0x40024200

Set bits in this register to allow this module to generate the corresponding interrupt.

**Table 439: INTEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WDT |

**Table 440: INTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:1 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |

**Table 440: INTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 0 | WDT | 0x0 | RW | Watchdog Timer Interrupt. |

### 9.2.2.5 *INTSTAT Register*

**WDT Interrupt register: Status**

**OFFSET:** 0x00000204

**INSTANCE 0 ADDRESS:** 0x40024204

Read bits from this register to discover the cause of a recent interrupt.

**Table 441: INTSTAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WDT |

**Table 442: INTSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:1 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 0 | WDT | 0x0 | RW | Watchdog Timer Interrupt. |

### 9.2.2.6 *INTCLR Register*

**WDT Interrupt register: Clear**

**OFFSET:** 0x00000208

**INSTANCE 0 ADDRESS:** 0x40024208

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

**Table 443: INTCLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WDT |

Table 444: INTCLR Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:1 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 0 | WDT | 0x0 | RW | Watchdog Timer Interrupt. |

### *9.2.2.7 INTSET Register*

**WDT Interrupt register: Set**

**OFFSET:** 0x0000020C

**INSTANCE 0 ADDRESS:** 0x4002420C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

Table 445: INTSET Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WDT |

Table 446: INTSET Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:1 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 0 | WDT | 0x0 | RW | Watchdog Timer Interrupt. |

# 10. Reset Generator Module

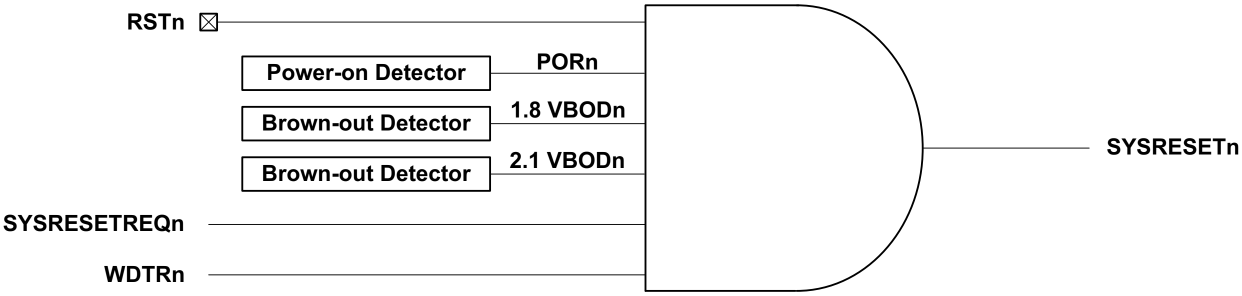


**Figure 46. Block diagram for the Reset Generator Module**

## 10.1 Functional Overview

The Reset Generator Module (RSTGEN) monitors a variety of reset signals and asserts the active low system reset (SYSRESETn) accordingly. A reset causes the entire system to be re-initialized, and the cause of the most recent reset is indicated by the STAT register.

Reset sources are described in the subsequent sections and include:

- External reset pin (RSTn)
- Power-on event
- Brown-out event
- Software request (SYSRESETREQn)
- Watchdog expiration

## 10.2 External Reset Pin

The active-low RSTn pin can be used to generate a reset using an off-chip component (e.g., a push-button). An internal pull-up resistor in the RSTn pad enables optional floating of the RSTn pin, and a de-bounce circuit ensures that bounce glitches on RSTn does not cause unintentional resets. The RSTn pin is not maskable. An internal pull-down device will be active during a brownout event pulling the RSTn pin low. See Figure 47.

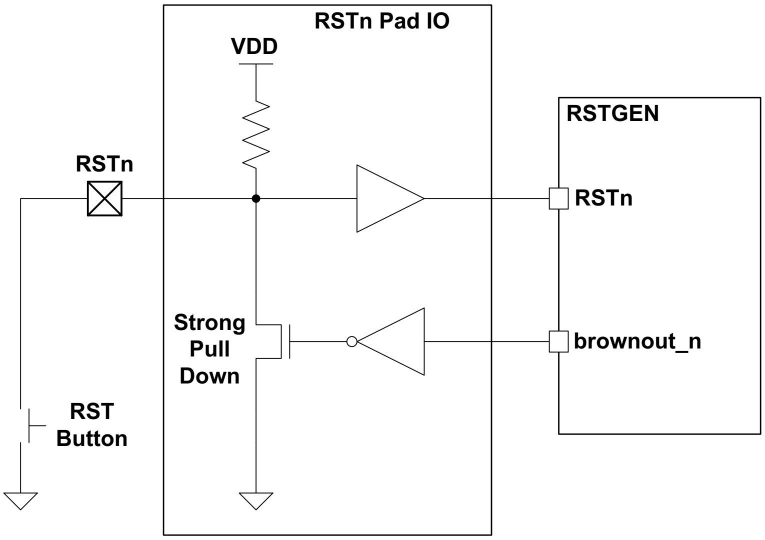


**Figure 47. Block diagram of circuitry for Reset pin**

## 10.3 Power-on Event

An integrated power-on detector monitors the supply voltage and keeps SYSRESETn asserted while VDD is below the rising power-on voltage, $V_{POR+}$ (2.05 V). When VDD rises above $V_{POR}$ at initial power on, the reset module will initialize the low power analog circuitry followed by de-assertion of SYSRESETn, and normal operation proceeds. SYSRESETn is re-asserted as soon as VDD falls below the falling power-on voltage, $V_{POR-}$ (1.75 V). The power-on reset signal, PORn, is not maskable.

## 10.4 Brown-out Event

An integrated brown-out detector monitors the supply voltage and causes an automatic and nonconfigurable reset when the voltage has fallen below the 1.8 V threshold. An optional reset or interrupt can be enabled when the brown-out detector indicates the supply voltage has fallen below the 2.1 V threshold.

In the event the supply voltage falls below the 1.8 V threshold, or 2.1 V threshold if enabled, the reset module will initiate a system reset, enabling the RSTn pull-down and driving the reset pin low. A 1.8 V or 2.1 V BOD reset will be reflected by the setting of the BORSTAT bit in the RSTGEN's STAT Register after reset.

## 10.5 Software Reset

A reset may be generated via software using the Application Interrupt and Reset Control Register (AIRCR) defined in the Cortex-M4. For additional information on the AIRCR, see the Arm document titled “Cortex-M4 Devices Generic User Guide.” The software reset request is not maskable.

## 10.6 Watchdog Reset

The Watchdog Timer sub-module generates an interrupt if it has not been properly managed by software within a pre-defined time. The watchdog reset is maskable.

## 10.7 RSTGEN Registers

**MCU Reset Generator**

**INSTANCE 0 BASE ADDRESS:**0x40000000

### *10.7.1 Register Memory Map*

**Table 447: RSTGEN Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0x40000000 | CFG | Configuration Register |
| 0x40000004 | SWPOI | Software POI Reset |
| 0x40000008 | SWPOR | Software POR Reset |
| 0x4000000C | STAT | Status Register |
| 0x40000010 | CLRSTAT | Clear the status register |
| 0x40000200 | INTEN | Reset Interrupt register: Enable |
| 0x40000204 | INTSTAT | Reset Interrupt register: Status |
| 0x40000208 | INTCLR | Reset Interrupt register: Clear |
| 0x4000020C | INTSET | Reset Interrupt register: Set |

### *10.7.2 RSTGEN Registers*

### *10.7.3 CFG Register*

**Configuration Register**

**OFFSET:** 0x00000000

**INSTANCE 0 ADDRESS:** 0x40000000

Configuration Register

**Table 448: CFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | WDREN | BODHREN |

**Table 449: CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:2 | RSVD | 0x0 | RO | RESERVED. |
| 1 | WDREN | 0x0 | RW | Watchdog Timer Reset Enable.  NOTE: The WDT module must also be configured for WDT reset. |
| 0 | BODHREN | 0x0 | RW | Brown out high (2.1v) reset enable. |

### 10.7.3.1 *SWPOI Register*

**Software POI Reset**

**OFFSET:** 0x00000004

**INSTANCE 0 ADDRESS:** 0x40000004

Software POI Reset

**Table 450: SWPOI Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="24">RSVD</td><td colspan="8">SWPOIKEY</td></tr>
</table>

**Table 451: SWPOI Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED. |
| 7:0 | SWPOIKEY | 0x0 | WO | 0x1B generates a software POI reset.<br>KEYVALUE = 0x1B - Writing 0x1B key value generates a software POI reset. |

### 10.7.3.2 *SWPOR Register*

**Software POR Reset**

**OFFSET:** 0x00000008

**INSTANCE 0 ADDRESS:** 0x40000008

Software POR Reset

**Table 452: SWPOR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | SWPORKEY | | | | | | | |

**Table 453: SWPOR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED. |
| 7:0 | SWPORKEY | 0x0 | WO | 0xD4 generates a software POR reset.<br>KEYVALUE = 0xD4 - Writing 0xD4 key value generates a software POR reset. |

### 10.7.3.3 *STAT Register*

**Status Register**

**OFFSET:** 0x0000000C

**INSTANCE 0 ADDRESS:** 0x4000000C

Status Register

**Table 454: STAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | WDRSTAT | DBGRSTAT | POIRSTAT | SWRSTAT | BORSTAT | PORSTAT | EXRSTAT |

**Table 455: STAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:7 | RSVD | 0x0 | RO | RESERVED. |

**Table 455: STAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 6 | WDRSTAT | 0x0 | RO | Reset was initiated by a Watchdog Timer Reset. |
| 5 | DBGRSTAT | 0x0 | RO | Reset was a initiated by Debugger Reset. |
| 4 | POIRSTAT | 0x0 | RO | Reset was a initiated by Software POI Reset. |
| 3 | SWRSTAT | 0x0 | RO | Reset was a initiated by SW POR or AIRCR Reset. |
| 2 | BORSTAT | 0x0 | RO | Reset was initiated by a Brown-Out Reset. |
| 1 | PORSTAT | 0x0 | RO | Reset was initiated by a Power-On Reset. |
| 0 | EXRSTAT | 0x0 | RO | Reset was initiated by an External Reset. |

### *10.7.3.4 CLRSTAT Register*

**Clear the status register**

**OFFSET:** 0x00000010

**INSTANCE 0 ADDRESS:** 0x40000010

Clear the status register

**Table 456: CLRSTAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | CLRSTAT |

**Table 457: CLRSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:1 | RSVD | 0x0 | RO | RESERVED. |
| 0 | CLRSTAT | 0x0 | WO | Writing a 1 to this bit clears all bits in the RST_STAT. |

### 10.7.3.5 INTEN Register

**Reset Interrupt register: Enable**

**OFFSET:** 0x00000200

**INSTANCE 0 ADDRESS:** 0x40000200

Set bits in this register to allow this module to generate the corresponding interrupt.

**Table 458: INTEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | BODH |

**Table 459: INTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:1 | RSVD | 0x0 | RO | RESERVED. |
| 0 | BODH | 0x0 | RW | Enables an interrupt that triggers when VCC is below BODH level. |

### 10.7.3.6 INTSTAT Register

**Reset Interrupt register: Status**

**OFFSET:** 0x00000204

**INSTANCE 0 ADDRESS:** 0x40000204

Read bits from this register to discover the cause of a recent interrupt.

**Table 460: INTSTAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | BODH |

**Table 461: INTSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:1 | RSVD | 0x0 | RO | RESERVED. |

**Table 461: INTSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 0 | BODH | 0x0 | RW | Enables an interrupt that triggers when VCC is below BODH level. |

### 10.7.3.7 *INTCLR Register*

**Reset Interrupt register: Clear**

**OFFSET:** 0x00000208

**INSTANCE 0 ADDRESS:** 0x40000208

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

**Table 462: INTCLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | BODH |

**Table 463: INTCLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:1 | RSVD | 0x0 | RO | RESERVED. |
| 0 | BODH | 0x0 | RW | Enables an interrupt that triggers when VCC is below BODH level. |

### 10.7.3.8 *INTSET Register*

**Reset Interrupt register: Set**

**OFFSET:** 0x0000020C

**INSTANCE 0 ADDRESS:** 0x4000020C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

**Table 464: INTSET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | BODH |

**Table 465: INTSET Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:1 | RSVD | 0x0 | RO | RESERVED. |
| 0 | BODH | 0x0 | RW | Enables an interrupt that triggers when VCC is below BODH level. |

# 11. UART Module

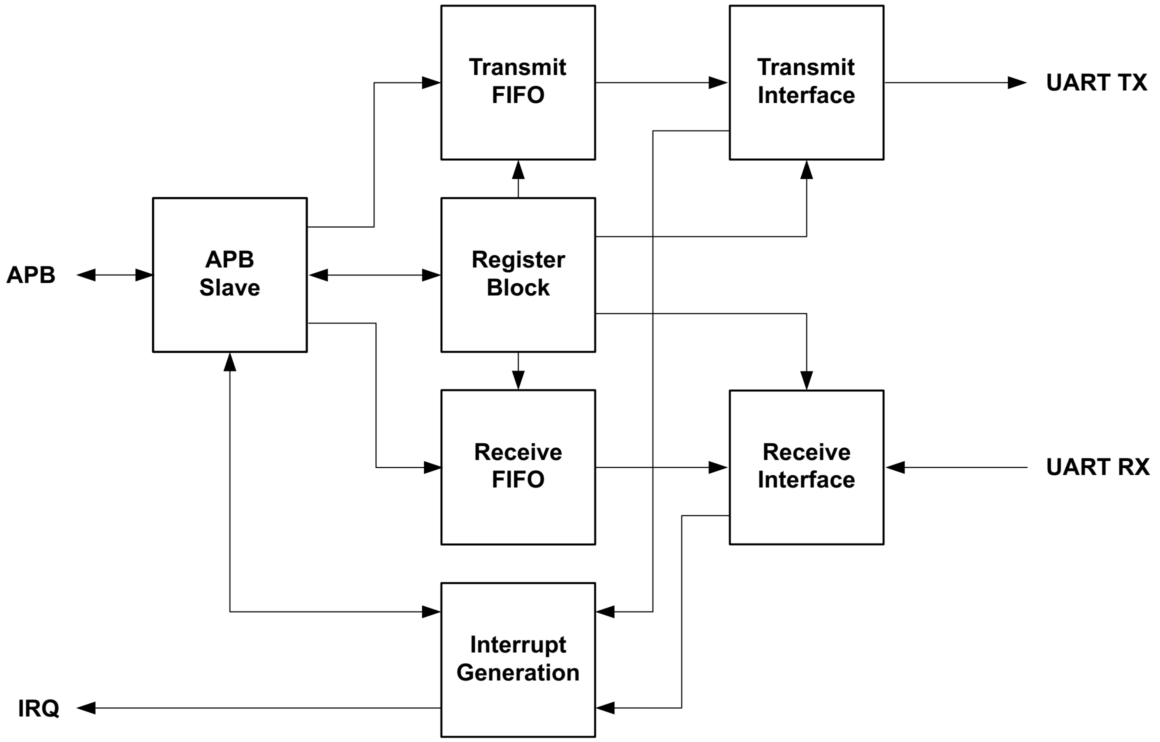


**Figure 48. Block Diagram for the UART Module**

## 11.1 Features

The UART Module includes the following key features:

- Operates independently, allowing the SoC to enter a low power sleep mode during communication
- 32 x 8 transmit FIFO and 32 x 12 receive FIFO to reduce MCU computational load
- Programmable baud rate generator
- Fully programmable data size, parity, and stop bit length
- Programmable hardware flow control
- Support for full-duplex and half-duplex communication
- Loop back functionality for diagnostics and testing

## 11.2 Functional Overview

Shown in Figure 48, the UART Module converts parallel data written through the APB Slave port into serial data which is transmitted to an external device. It also receives serial data from an external device and converts it to parallel data, which is then stored in a buffer until the CPU reads the data.

The UART Module includes a programmable baud rate generator. An interrupt generator will optionally send interrupts to the CPU core for transmit, receive and error events.

Internally, the UART Module maintains two FIFOs. The transmit FIFO is 1-byte wide with 32 locations. The receive FIFO is 12-bits wide with 32 locations. The extra four bits in the receive FIFO are used to capture any error status information that the MCU needs to analyze.

Clocking to the UART serial logic is generated by a dedicated UARTCLK from the Clock Generator Module. The frequency of this clock is determined by the desired baud rate. For maximum baud rates, this clock would be clocked at the 24 MHz maximum as generated by the HFRC.

The major functional blocks of the UART are discussed briefly in the subsequent sections.

## 11.3 Enabling and Selecting the UART Clock

Before the UART Module can communicate, a clock frequency must be selected for the UART clock, and the clock must then be enabled. Use the UART_CR register to select the desired clock frequency, $F_{UART}$. After selecting the desired frequency, enable the UART clock using the CLK_GEN_UARTEN register. Unlike other modules that automatically turn off their clock sources automatically, the UART clock must be manually disabled using the aforementioned CLK_GEN_UARTEN register. To ensure minimum energy operation, the UART clock should be enabled for the minimum time possible and should be disabled as soon as UART communication is complete.

## 11.4 Configuration

The UART Register Block in Figure 48 may be set to configure the UART Module. The data width, number of stop bits, and parity may all be configured using the UART_LCRH register.

The baud rate is configured using the integer UART_IBRD and UART_FBRD registers. The correct values for UART_IBRD and UART_FBRD may be determined according to the following equation:

$F_{UART}/(16\cdot BR) = IBRD + FBRD$

$F_{UART}$ is the frequency of the UART clock. BR is the desired baud rate. IBRD is the integer portion of the baud rate divisor. FBRD is the fractional portion of the baud rate divisor.

The UART Module supports independent CTS and RTS hardware flow control. All flow control configuration may be set using the UART_CR register.

## 11.5 Transmit FIFO and Receive FIFO

The transmit and receive FIFOs may both be accessed via the same 8-bit word in the UART_DR register. The transmit FIFO stores up to 32 8-bit words and can be written using writes to UART_DR. The receive FIFO stores up to 32 12-bit words and can be read using reads to UART_DR. Note that each 12-bit receive FIFO word includes an 8-bit data word and a 4-bit error status word.

## 11.6 UART Registers

**Serial UART**

**INSTANCE 0 BASE ADDRESS:**0x4001C000

### *11.6.1 Register Memory Map*

**Table 466: UART Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0x4001C000 | DR | UART Data Register |
| 0x4001C004 | RSR | UART Status Register |
| 0x4001C018 | FR | Flag Register |
| 0x4001C020 | ILPR | IrDA Counter |
| 0x4001C024 | IBRD | Integer Baud Rate Divisor |
| 0x4001C028 | FBRD | Fractional Baud Rate Divisor |
| 0x4001C02C | LCRH | Line Control High |
| 0x4001C030 | CR | Control Register |
| 0x4001C034 | IFLS | FIFO Interrupt Level Select |
| 0x4001C038 | IER | Interrupt Enable |
| 0x4001C03C | IES | Interrupt Status |
| 0x4001C040 | MIS | Masked Interrupt Status |
| 0x4001C044 | IEC | Interrupt Clear |

### *11.6.2 UART Registers*

#### *11.6.2.1 DR Register*

**UART Data Register**

**OFFSET:** 0x00000000

**INSTANCE 0 ADDRESS:** 0x4001C000

UART Data Register

**Table 467: DR Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="20">RSVD</td><td>OEDATA</td><td>BEDATA</td><td>PEDATA</td><td>FEDATA</td><td colspan="8">DATA</td></tr>
</table>

Table 468: DR Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:12 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 11 | OEDATA | 0x0 | RO | This is the overrun error indicator.<br>NOERR = 0x0 - No error on UART OEDATA, overrun error indicator.<br>ERR = 0x1 - Error on UART OEDATA, overrun error indicator. |
| 10 | BEDATA | 0x0 | RO | This is the break error indicator.<br>NOERR = 0x0 - No error on UART BEDATA, break error indicator.<br>ERR = 0x1 - Error on UART BEDATA, break error indicator. |
| 9 | PEDATA | 0x0 | RO | This is the parity error indicator.<br>NOERR = 0x0 - No error on UART PEDATA, parity error indicator.<br>ERR = 0x1 - Error on UART PEDATA, parity error indicator. |
| 8 | FEDATA | 0x0 | RO | This is the framing error indicator.<br>NOERR = 0x0 - No error on UART FEDATA, framing error indicator.<br>ERR = 0x1 - Error on UART FEDATA, framing error indicator. |
| 7:0 | DATA | 0x0 | RW | This is the UART data port. |

### *11.6.2.2 RSR Register*

**UART Status Register**

**OFFSET:** 0x00000004

**INSTANCE 0 ADDRESS:** 0x4001C004

UART Status Register

Table 469: RSR Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | OESTAT | BESTAT | PESTAT | FESTAT |

Table 470: RSR Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:4 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |

**Table 470: RSR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 3 | OESTAT | 0x0 | RW | This is the overrun error indicator.<br>NOERR = 0x0 - No error on UART OESTAT, overrun error indicator.<br>ERR = 0x1 - Error on UART OESTAT, overrun error indicator. |
| 2 | BESTAT | 0x0 | RW | This is the break error indicator.<br>NOERR = 0x0 - No error on UART BESTAT, break error indicator.<br>ERR = 0x1 - Error on UART BESTAT, break error indicator. |
| 1 | PESTAT | 0x0 | RW | This is the parity error indicator.<br>NOERR = 0x0 - No error on UART PESTAT, parity error indicator.<br>ERR = 0x1 - Error on UART PESTAT, parity error indicator. |
| 0 | FESTAT | 0x0 | RW | This is the framing error indicator.<br>NOERR = 0x0 - No error on UART FESTAT, framing error indicator.<br>ERR = 0x1 - Error on UART FESTAT, framing error indicator. |

### *11.6.2.3 FR Register*

**Flag Register**

**OFFSET:** 0x00000018

**INSTANCE 0 ADDRESS:** 0x4001C018

Flag Register

**Table 471: FR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | RI | TXFE | RXFF | TXFF | RXFE | BUSY | DCD | DSR | CTS |

**Table 472: FR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:9 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 8 | RI | 0x0 | RO | This bit holds the ring indicator. |
| 7 | TXFE | 0x0 | RO | This bit holds the transmit FIFO empty indicator.<br>XMTFIFO_EMPTY = 0x1 - Transmit fifo is empty. |
| 6 | RXFF | 0x0 | RO | This bit holds the receive FIFO full indicator.<br>RCVFIFO_FULL = 0x1 - Receive fifo is full. |

**Table 472: FR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 5 | TXFF | 0x0 | RO | This bit holds the transmit FIFO full indicator.<br>XMTFIFO_FULL = 0x1 - Transmit fifo is full. |
| 4 | RXFE | 0x0 | RO | This bit holds the receive FIFO empty indicator.<br>RCVFIFO_EMPTY = 0x1 - Receive fifo is empty. |
| 3 | BUSY | 0x0 | RO | This bit holds the busy indicator.<br>BUSY = 0x1 - UART busy indicator. |
| 2 | DCD | 0x0 | RO | This bit holds the data carrier detect indicator.<br>DETECTED = 0x1 - Data carrier detect detected. |
| 1 | DSR | 0x0 | RO | This bit holds the data set ready indicator.<br>READY = 0x1 - Data set ready. |
| 0 | CTS | 0x0 | RO | This bit holds the clear to send indicator.<br>CLEARTOSEND = 0x1 - Clear to send is indicated. |

### *11.6.2.4 ILPR Register*

**IrDA Counter**

**OFFSET:** 0x00000020

**INSTANCE 0 ADDRESS:** 0x4001C020

IrDA Counter

**Table 473: ILPR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | ILPDVSR | | | | | | | |

**Table 474: ILPR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 7:0 | ILPDVSR | 0x0 | RW | These bits hold the IrDA counter divisor. |

### 11.6.2.5 *IBRD Register*

**Integer Baud Rate Divisor**

**OFFSET:** 0x00000024

**INSTANCE 0 ADDRESS:** 0x4001C024

Integer Baud Rate Divisor

**Table 475: IBRD Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | DIVINT | | | | | | | | | | | | | | | |

**Table 476: IBRD Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 15:0 | DIVINT | 0x0 | RW | These bits hold the baud integer divisor. |

### 11.6.2.6 *FBRD Register*

**Fractional Baud Rate Divisor**

**OFFSET:** 0x00000028

**INSTANCE 0 ADDRESS:** 0x4001C028

Fractional Baud Rate Divisor

**Table 477: FBRD Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | DIVFRAC | | | | | |

**Table 478: FBRD Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:6 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 5:0 | DIVFRAC | 0x0 | RW | These bits hold the baud fractional divisor. |

### *11.6.2.7 LCRH Register*

**Line Control High**

**OFFSET:** 0x0000002C

**INSTANCE 0 ADDRESS:** 0x4001C02C

Line Control High

**Table 479: LCRH Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | SPS | WLEN | | FEN | STP2 | EPS | PEN | BRK |

**Table 480: LCRH Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 7 | SPS | 0x0 | RW | This bit holds the stick parity select. |
| 6:5 | WLEN | 0x0 | RW | These bits hold the write length. |
| 4 | FEN | 0x0 | RW | This bit holds the FIFO enable. |
| 3 | STP2 | 0x0 | RW | This bit holds the two stop bits select. |
| 2 | EPS | 0x0 | RW | This bit holds the even parity select. |
| 1 | PEN | 0x0 | RW | This bit holds the parity enable. |
| 0 | BRK | 0x0 | RW | This bit holds the break set. |

### *11.6.2.8 CR Register*

**Control Register**

**OFFSET:** 0x00000030

**INSTANCE 0 ADDRESS:** 0x4001C030

Control Register

**Table 481: CR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | CTSEN | RTSEN | OUT2 | OUT1 | RTS | DTR | RXE | TXE | LBE | CLKSEL | | | CLKEN | SIRLP | SIREN | UARTEN |

**Table 482: CR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 15 | CTSEN | 0x0 | RW | This bit enables CTS hardware flow control. |
| 14 | RTSEN | 0x0 | RW | This bit enables RTS hardware flow control. |
| 13 | OUT2 | 0x0 | RW | This bit holds modem Out2. |
| 12 | OUT1 | 0x0 | RW | This bit holds modem Out1. |
| 11 | RTS | 0x0 | RW | This bit enables request to send. |
| 10 | DTR | 0x0 | RW | This bit enables data transmit ready. |
| 9 | RXE | 0x1 | RW | This bit is the receive enable. |
| 8 | TXE | 0x1 | RW | This bit is the transmit enable. |
| 7 | LBE | 0x0 | RW | This bit is the loopback enable. |

Table 482: CR Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 6:4 | CLKSEL | 0x0 | RW | This bitfield is the UART clock select.<br><br>NOCLK = 0x0 - No UART clock.  This is the low power default.<br>24MHZ = 0x1 - 24 MHz clock.  Must be used if CLKGEN CORESEL=0.<br>12MHZ = 0x2 - 12 MHz clock.  Must be used if CLKGEN CORESEL=1. Note that CORESEL=1 is unsupported by the IO Master.<br>6MHZ = 0x3 - 6 MHz clock.  Must be used if CLKGEN CORESEL=2, 3, or 4. Note that CORESEL=2 is unsupported.<br>3MHZ = 0x4 - 3 MHz clock.  Must be used if CLKGEN CORESEL=5, 6, or 7.<br>RSVD5 = 0x5 - Reserved.<br>RSVD6 = 0x6 - Reserved.<br>RSVD7 = 0x7 - Reserved. |
| 3 | CLKEN | 0x0 | RW | This bit is the UART clock enable. |
| 2 | SIRLP | 0x0 | RW | This bit is the SIR low power select. |
| 1 | SIREN | 0x0 | RW | This bit is the SIR ENDEC enable. |
| 0 | UARTEN | 0x0 | RW | This bit is the UART enable. |

### *11.6.2.9 IFLS Register*

**FIFO Interrupt Level Select**

**OFFSET:** 0x00000034

**INSTANCE 0 ADDRESS:** 0x4001C034

FIFO Interrupt Level Select

Table 483: IFLS Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | RXIFLSEL | | | TXIFLSEL | | |

Table 484: IFLS Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:6 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 5:3 | RXIFLSEL | 0x2 | RW | These bits hold the receive FIFO interrupt level. |
| 2:0 | TXIFLSEL | 0x2 | RW | These bits hold the transmit FIFO interrupt level. |

### *11.6.2.10IER Register*

**Interrupt Enable**

**OFFSET:** 0x00000038

**INSTANCE 0 ADDRESS:** 0x4001C038

Interrupt Enable

**Table 485: IER Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | OEIM | BEIM | PEIM | FEIM | RTIM | TXIM | RXIM | DSRMIM | DCDMIM | CTSMIM | RIMIM |

**Table 486: IER Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:11 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 10 | OEIM | 0x0 | RW | This bit holds the overflow interrupt enable. |
| 9 | BEIM | 0x0 | RW | This bit holds the break error interrupt enable. |
| 8 | PEIM | 0x0 | RW | This bit holds the parity error interrupt enable. |
| 7 | FEIM | 0x0 | RW | This bit holds the framing error interrupt enable. |
| 6 | RTIM | 0x0 | RW | This bit holds the receive timeout interrupt enable. |
| 5 | TXIM | 0x0 | RW | This bit holds the transmit interrupt enable. |
| 4 | RXIM | 0x0 | RW | This bit holds the receive interrupt enable. |
| 3 | DSRMIM | 0x0 | RW | This bit holds the modem DSR interrupt enable. |
| 2 | DCDMIM | 0x0 | RW | This bit holds the modem DCD interrupt enable. |
| 1 | CTSMIM | 0x0 | RW | This bit holds the modem CTS interrupt enable. |
| 0 | RIMIM | 0x0 | RW | This bit holds the modem RI interrupt enable. |

### *11.6.2.11 IES Register*

**Interrupt Status**

**OFFSET:** 0x0000003C

**INSTANCE 0 ADDRESS:** 0x4001C03C

Interrupt Status

**Table 487: IES Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | OERIS | BERIS | PERIS | FERIS | RTRIS | TXRIS | RXRIS | DSRMRIS | DCDMRIS | CTSMRIS | RIMRIS |

**Table 488: IES Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:11 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 10 | OERIS | 0x0 | RO | This bit holds the overflow interrupt status. |
| 9 | BERIS | 0x0 | RO | This bit holds the break error interrupt status. |
| 8 | PERIS | 0x0 | RO | This bit holds the parity error interrupt status. |
| 7 | FERIS | 0x0 | RO | This bit holds the framing error interrupt status. |
| 6 | RTRIS | 0x0 | RO | This bit holds the receive timeout interrupt status. |
| 5 | TXRIS | 0x0 | RO | This bit holds the transmit interrupt status. |
| 4 | RXRIS | 0x0 | RO | This bit holds the receive interrupt status. |
| 3 | DSRMRIS | 0x0 | RO | This bit holds the modem DSR interrupt status. |
| 2 | DCDMRIS | 0x0 | RO | This bit holds the modem DCD interrupt status. |
| 1 | CTSMRIS | 0x0 | RO | This bit holds the modem CTS interrupt status. |
| 0 | RIMRIS | 0x0 | RO | This bit holds the modem RI interrupt status. |

### *11.6.2.12MIS Register*

**Masked Interrupt Status**

**OFFSET:** 0x00000040

**INSTANCE 0 ADDRESS:** 0x4001C040

Masked Interrupt Status

**Table 489: MIS Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | OEMIS | BEMIS | PEMIS | FEMIS | RTMIS | TXMIS | RXMIS | DSRMMIS | DCDMMIS | CTSMMIS | RIMMIS |

**Table 490: MIS Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:11 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 10 | OEMIS | 0x0 | RO | This bit holds the overflow interrupt status masked. |
| 9 | BEMIS | 0x0 | RO | This bit holds the break error interrupt status masked. |
| 8 | PEMIS | 0x0 | RO | This bit holds the parity error interrupt status masked. |
| 7 | FEMIS | 0x0 | RO | This bit holds the framing error interrupt status masked. |
| 6 | RTMIS | 0x0 | RO | This bit holds the receive timeout interrupt status masked. |
| 5 | TXMIS | 0x0 | RO | This bit holds the transmit interrupt status masked. |
| 4 | RXMIS | 0x0 | RO | This bit holds the receive interrupt status masked. |
| 3 | DSRMMIS | 0x0 | RO | This bit holds the modem DSR interrupt status masked. |
| 2 | DCDMMIS | 0x0 | RO | This bit holds the modem DCD interrupt status masked. |
| 1 | CTSMMIS | 0x0 | RO | This bit holds the modem CTS interrupt status masked. |
| 0 | RIMMIS | 0x0 | RO | This bit holds the modem RI interrupt status masked. |

### *11.6.2.13IEC Register*

**Interrupt Clear**

**OFFSET:** 0x00000044

**INSTANCE 0 ADDRESS:** 0x4001C044

Interrupt Clear

**Table 491: IEC Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | OEIC | BEIC | PEIC | FEIC | RTIC | TXIC | RXIC | DSRMIC | DCDMIC | CTSMIC | RIMIC |

**Table 492: IEC Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:11 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 10 | OEIC | 0x0 | WO | This bit holds the overflow interrupt clear. |
| 9 | BEIC | 0x0 | WO | This bit holds the break error interrupt clear. |
| 8 | PEIC | 0x0 | WO | This bit holds the parity error interrupt clear. |
| 7 | FEIC | 0x0 | WO | This bit holds the framing error interrupt clear. |
| 6 | RTIC | 0x0 | WO | This bit holds the receive timeout interrupt clear. |
| 5 | TXIC | 0x0 | WO | This bit holds the transmit interrupt clear. |
| 4 | RXIC | 0x0 | WO | This bit holds the receive interrupt clear. |
| 3 | DSRMIC | 0x0 | WO | This bit holds the modem DSR interrupt clear. |
| 2 | DCDMIC | 0x0 | WO | This bit holds the modem DCD interrupt clear. |
| 1 | CTSMIC | 0x0 | WO | This bit holds the modem CTS interrupt clear. |
| 0 | RIMIC | 0x0 | WO | This bit holds the modem RI interrupt clear. |

# 12. ADC and Temperature Sensor Module

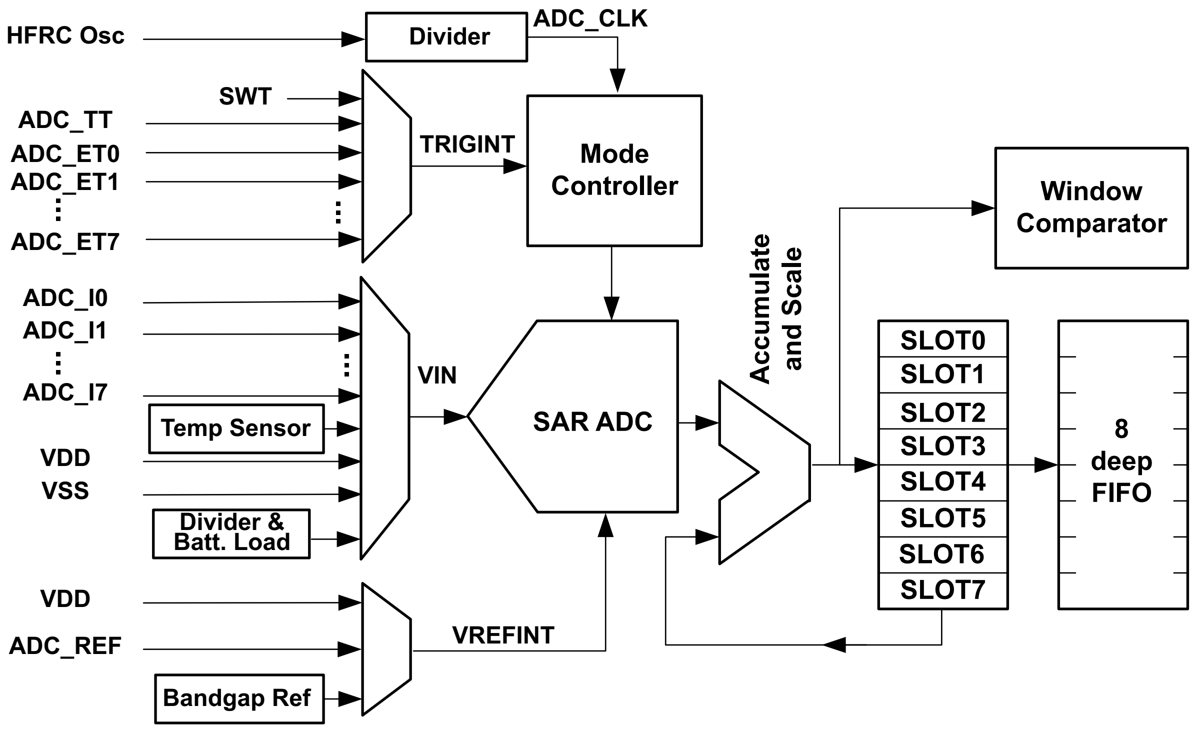


**Figure 49. Block Diagram for ADC and Temperature Sensor**

## 12.1 Features

The Analog-to-Digital Converter (ADC) and Temperature Sensor Module includes a single-ended 10-bit multi-channel Successive Approximation Register (SAR) ADC as shown in Figure 49.

Key features include:

- 12 user-selectable channels including:
  - External pins
    - 8 single ended external pins
  - Internal voltage (VSS)
  - Voltage divider (battery)
  - Temperature sensor
- User-selectable on-chip and off-chip reference voltages
- Single shot, repeating single shot, scan, and repeating scan modes
- User-selectable clock source for variable sampling rates
- User-selectable track/hold time
- Multiple conversion triggers including external pins, a timer, and a software write
- Automatically accumulate and scale module for hardware averaging of samples
- An 8-entry FIFO for storing measurement results and maximizing MCU sleep time
- Window comparator for monitoring voltages excursions into or out of user-selectable thresholds
- Multiple low power modes between conversions
- Up to 800 kSps sampling rate

## 12.2 Functional Overview

The Apollo SoC integrates a sophisticated Analog to Digital Converter (ADC) block for sensing both internal and external voltages. The block provides eight separately managed conversion requests, called slots. The result of each conversion requests is delivered to an eight deep FIFO. Firmware can utilize various interrupt notifications to determine when to collect the sampled data from the FIFO. This block is extremely effective at automatically managing its power states and its clock sources.

### *12.2.1 Clock Source and Dividers*

The ADC runs off of the HFRC clock source. When the ADC block is enabled, it requests an HFRC clock source. There is an automatic hardware hand shake between the clock generator and the ADC. If the ADC is the only block requesting an HFRC based clock then the HFRC will be automatically started. The ADC can select one of the following clock divisions from the 24 MHz HFRC:

- 12 MHz
- 6 MHz
- 3 MHz
- 1.5 MHz

### *12.2.2 12 Channel Analog Mux*

As shown in Figure 49, the ADC block contains a 12 channel analog multiplexer on the input port to the analog to digital converter. Eight of the GPIO pins on the Apollo SoC can be selected as analog inputs to the ADC through a combination of settings in the PAD configuration registers in the GPIO block and settings in the configuration registers described below.

The analog mux channels are connected as follows:

1. ADC_EXT0 external GPIO pin connection.
2. ADC_EXT1 external GPIO pin connection.
3. ADC_EXT2 external GPIO pin connection.
4. ADC_EXT3 external GPIO pin connection.
5. ADC_EXT4 external GPIO pin connection.
6. ADC_EXT5 external GPIO pin connection.
7. ADC_EXT6 external GPIO pin connection.
8. ADC_EXT7 external GPIO pin connection.
9. ADC_TEMP internal temperature sensor.
10. ADC_VDD internal connection to input power rails.
11. ADC_VSS internal ground connection.
12. ADC_DIV3_LOAD internal voltage divide by 3 connection to the input power rail with the 500 ohm load resistor turned on.

Refer to the detailed register information below for the exact coding of the channel selection bit field. Also the use of the voltage divider and switchable load resistor are detailed below. See “ADC State Diagram” on page 322.

### *12.2.3 Triggering and Trigger Sources*

The ADC block can be initially triggered from one of eight sources. Once triggered, it can be repetitively triggered from counter/timer number three (3). Seven of the GPIO pins on the Apollo SoC can be selected as trigger inputs to the ADC through a combination of settings in the PAD configuration registers in the GPIO block and settings in SLOT configuration registers described below. Initial trigger sources are as follows:

0. ADC_TRIG0 external GPIO pin connection.
1. ADC_TRIG1 external GPIO pin connection.
2. ADC_TRIG2 external GPIO pin connection.
3. ADC_TRIG3 external GPIO pin connection.
4. ADC_TRIG5 external GPIO pin connection.
5. ADC_TRIG6 external GPIO pin connection.
6. ADC_TRIG7 external GPIO pin connection.
7. ADC_SWT software trigger.

**NOTE**: Although ADC External Trigger 4 (TRIG4) is shown as a valid pin function selection elsewhere in the datasheet (“Apollo Pad Function Mapping” in the GPIO chapter, “Pad Configuration Functions” on page 140, Table 246, “ADC Trigger Input Configuration,” on page 159, and PADREGK Register Bits table), it should not be used as an ADC external trigger.

Refer to the ADC Configuration Register in the detailed register information section below. The initial trigger source is selected in the TRIGSEL field, as shown below. In addition, one can select a trigger polarity in this register. A number of GPIO pin trigger sources are provided to allow pin configuration flexibility at the system definition and board layout phases of development.

The software trigger is effected by writing 0x37 to the software trigger register in the ADC block. Note that writing 0x37 to the software trigger register will initiate a scan regardless of which trigger source is selected. However, a hardware trigger source will not initiate a scan if the software trigger has been selected.

The discussion of the use of counter/timer three as a source for repetitive triggering is deferred until later in this chapter. See “Repeat Mode” on page 321.

NOTE

A trigger event applies to all enabled slots as a whole. Individual slots can not be separately triggered.

## 12.3 Voltage Reference Sources

The Apollo SoC ADC supports one of three reference sources to be used for the analog to digital conversion step:

- Bandgap Reference Source
- VDD internal voltage rail
- External reference via GPIO16

### *12.3.1 Eight Automatically Managed Conversion Slots*

If the SAR is the heart of the ADC block, then the conversion SLOTs and their mode controller are the brains of this block. The ADC block contains eight conversion slot control registers, one for each of the eight slots. These can be thought of as time slots in the conversion process. When a slot is enabled, it participates in a conversion cycle. The ADC’s mode controller cycles through up to eight time slots each time it is triggered. For each slot that is enabled, a conversion cycle is performed based on the settings in the slot configuration register for that slot. Slots are enabled when the LSB of the slot configuration is set to one.See Table 493 on page 316. The window comparator enable will be discussed in a subsequent section, below. See “” on page 318. The number of samples to accumulate will also be explained in a subsequent section. See “Automatic Sample Accumulation and Scaling” on page 316.

**Table 493: One SLOT Configuration Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | # Samples to Accum. | | | Reserved | | | | | TRACK & HOLD | | | Reserved | | | | CHANNEL SELECT | | | | Reserved | | | | | | WINDOW_COMP | SLOT_ENABLE |

As described above, the channel select bit field specifies which one of the analog multiplexer channels will be used for the conversions requested for an individual slot. See “12 Channel Analog Mux” on page 314.

As its name implies, the track and hold bit field controls the time from when an analog mux channel is selected until the ADC hold circuit captures the analog value on that input. Refer to the detailed register section for the specifics of programming this bit field, but the bit field essentially offers powers of two increasing hold times from 1 to 128 ADC clocks. Note that this track and hold time can be specified independently for each of the eight conversion slots.

Each of the eight conversion slots can independently specify:

- Analog Multiplexer Channel Selection
- Track and Hold Time
- Participation in Window Comparisons
- Automatic Sample Accumulation

### *12.3.2 Automatic Sample Accumulation and Scaling*

The ADC block offers a facility for the automatic accumulation of samples without requiring core involvement. Thus up to 128 samples per slot can be accumulated without waking the core. This facilitates averaging algorithms to smooth out the data samples. Each slot can request from 1 to 128 samples to be accumulated before producing a result in the FIFO.

> NOTE
>
> Each slot can independently specify how many samples to accumulate so results can enter the FIFO from different slots at different rates.

All slots write their accumulated results to the FIFO in exactly the same format regardless of how many samples were accumulated to produce the results. Table 494 shows the format that is used by all conversions. This is a scaled integer format with a 6-bit fractional part. The format for **ALL** results is this 10.6 format.

**IMPORTANT:** if the accumulation control for a slot is set for one sample then the 10-bit value coming from the ADC will be inserted into bits 6 through 15 in this format.

**Table 494: 10.6 ADC Sample Format**

| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10-bit Integer | | | | | | | | | | 6-bit Fraction | | | | | |

Each slot contains a 17-bit accumulator as shown in Table 495. When the ADC is triggered for the last sample of an accumulation, the accumulator is cleared. Then when each active slot obtains a conversion from the analog side, it is added to the value in its accumulator.

If a slot is set to accumulate 128 samples per result then the accumulator could reach a maximum value of:

128*1023 = 128*1023 = 130944 = $2^{17}$ - 128, hence the 17 bit accumulator.

**Table 495: Per Slot Sample Accumulator**

| 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Accumulator | | | | | | | | | | | | | | | | |

Each completed accumulation must be scaled to obtain the required 10.6 format since **ALL** results must be reported in this 10.6 format. Each setting for the number of samples to accumulate bit field of the slot configuration register requires a different scale factor.

Table 496 shows the maximum possible accumulated values. Note that 64 sample accumulation produces a result that is exactly correct or the 10.6 format results so it is copied unscaled in to the FIFO. Furthermore, note that 128 sample accumulation can produce a result that is too large for the 10.6 format. These results are right shifted by one before they are written to the FIFO. All of the remaining sample accumulation settings must have their results left shifted to produce the desired 10.6 format.

Finally, note that for the 128 sample accumulation case, the LSB of the accumulator is discarded when the results are written to the FIFO.

Most importantly, note that for the 1 sample accumulation case, the 10-bit converter value is shifted left by six to produce the 10.6 format to write into the FIFO.

**Table 496: Accumulator Scaling**

| # Samples | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 128 | 10.6 | | | | | | | | | | | | | | | | 0 |
| 64 | X | 10.6 | | | | | | | | | | | | | | | |
| 32 | X | X | 10.5 | | | | | | | | | | | | | | |
| 16 | X | X | X | 10.4 | | | | | | | | | | | | | |
| 8 | X | X | X | X | 10.3 | | | | | | | | | | | | |
| 4 | X | X | X | X | X | 10.2 | | | | | | | | | | | |
| 2 | X | X | X | X | X | X | 10.1 | | | | | | | | | | |
| 1 | X | X | X | X | X | X | X | 10 | | | | | | | | | |

### 12.3.3 *Eight Entry Result FIFO*

All results written to the FIFO have exactly the same format as shown in Table 497. The properly scaled accumulation results are written the lower half word in the aforementioned 10.6 format. Since each slot can produce results at a different rate, the slot number generating the result is also written to the FIFO. Thus the interrupt handler servicing ADC interrupts can easily distribute results to different RTOS tasks by simply looking up the target task using the slot number from the FIFO register.

**Table 497: FIFO Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Reserved | | | | | Slot Number. | | | Reserved | | | | FIFO Count | | | | FIFO DATA | | | | | | | | | | | | | | | |

Two other features greatly simplify the task faced by firmware developers of interrupt service routines for the ADC block:

1. The FIFO count bit field is not really stored in the FIFO. Instead it is a live count of the number of valid entries currently residing in the FIFO. If the interrupt service routine was entered because of a conversion then this value will be at least one. When the interrupts routine is entered it can pull successive sample values from the FIFO until this bit field goes to zero. Thus avoiding wasteful re-entry of the interrupt service routine. Note that no further I/O bus read is required to determine the FIFO depth.
2. This FIFO has no read side effects. This is important to firmware for a number of reasons. One important result is that the FIFO register can be freely read repetitively by a debugger without affecting the state of the FIFO. In order to pop this FIFO and look at the next result, if any, one simply writes any value to this register. Any time the FIFO is read, then the compiler has gone to the trouble of generating an address for the read. To pop the FIFO, one simply writes to that same address with any value. This give firmware a positive handshake mechanism to control exactly when the FIFO pops.
3. When a conversion completes resulting in hardware populating the 6th valid FIFO entry, the FIFOOVR1 (FIFO 75% full) interrupt status bit will be set. When a conversion completes resulting in hardware populating the 8th valid FIFO entry, the FIFOOVR2 interrupt status bit will be set. In a FIFO full condition with 8 valid entries, the ADC will not overwrite existing valid FIFO contents. Before subsequent conversions will populate the FIFO with conversion data, software must free an open FIFO entry by writing to the FIFO Register or by resetting the ADC by disabling and enabling the ADC using the ADC_CFG register.

### 12.3.4 *Window Comparator*

A window comparator is provided which can generate an interrupt whenever a sample is determined to be inside the window limits or outside the window limits. These are two separate interrupts with separate interrupt enables. Thus one can request an interrupt any time a specified slot makes an excursion outside the window comparator limits.

Firmware has to participate in the determination of whether an actual excursion occurred. The window comparator interrupts set their corresponding interrupt status bits continuously whenever the inside or outside condition is true. Thus if one enables and receives an "*excursion*" interrupt then the status bit can't be usefully cleared while the ADC slot is sampling values outside the limits. That is, if one receives an excursion interrupt and clears the status bit, it will immediately set again if the next ADC sample is still outside the limits. Thus firmware should reconfigure the interrupt enables upon receiving an excursion interrupt so that the next interrupt will occur when an ADC sample ultimately goes back inside the window limits. Firmware may also want to change the windows comparator limit at that time to utilize a little hysteresis in these window comparator decisions.

**Table 498: Window Comparator Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Upper Limit | | | | | | | | | | | | | | | | Lower Limit | | | | | | | | | | | | | | | |

The determination of whether a sample is *inside* or *outside* of the window limits is made by comparing the 10.6 format of the slot result that will be written to the FIFO with the 16 bit window limits. An ADC sample is inside if the following relation is true:

Lower Limit <= ADC 10.6 SAMPLE <= Upper Limit

Thus setting both limits to the same value, say 700.0 (0x2BC<<6 = 0xAF00), will only produce an inside interrupt when the ADC sample is exactly 700.0 (0xAF00). Furthermore, note that if the lower limit is set to zero (0x0000) and the upper limit is set to 0xFFFF then all accumulated results from the ADC will be inside the window limits and no excursion interrupts can ever by generated. In fact, in this case, the incursion interrupt status bit will be set for every sample from any active slot with its window comparator bit enabled. If the incursion interrupt is enabled then an interrupt will be generated for every such sample written to the FIFO.

The window comparator limits are a shared resource and apply to all active slots which have their window comparator bits enabled.

## 12.4 Operating Modes and the Mode Controller

The mode controller of Figure 49 is a sophisticated state machine that manages not only the time slot conversions but also the power state of the ADC analog components and the hand shake with the clock generator to start the HFRC clock source if required. Thus once the various control registers are initialized, the core can go to sleep and only wake up when there are valid samples in the FIFO for the interrupt service routine to distribute. Firmware does not have to keep track of which block is using the HFRC clock source since the devices in conjunction with the clock generator manage this automatically. The ADC block’s mode controller participates in this clock management protocol.

From a firmware perspective, the ADC mode controller is controlled with bit fields in the ADC configuration register and from the various bit fields in the eight slot configuration registers.

The most over-riding control is the ADCEN bit in the ADC configuration register. Setting this bit to zero has many of the effects of a software reset, such as reseting the FIFO pointers. Setting this bit to one enables the mode controller to examine its inputs and proceed to autonomously handle analog to digital conversions.

An ADC scan is the process of sampling the analog voltages at each input of the SAR ADC of Figure 49 following a trigger event. If the ADC is enabled and one or more slots are enabled, a scan is initiated after the ADC receives a trigger through one of the configured trigger sources. The scan flowchart diagram can be found in Figure 50.

An ADC conversion is the process of averaging measurements following one or more scans for each slot that is enabled.

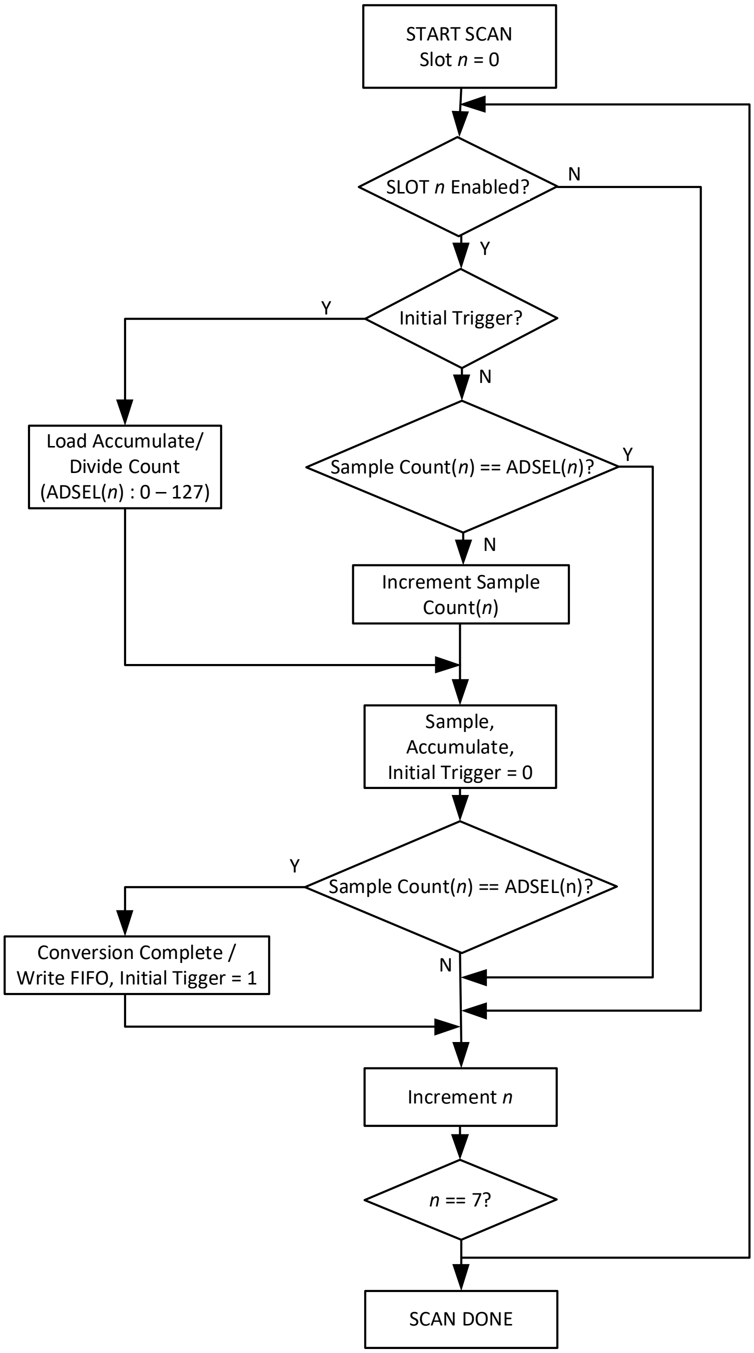


**Figure 50. Scan Flowchart**

### *12.4.1 Single Mode*

In single mode, one trigger event produces one scan of all enabled slots. Depending on the settings of the accumulate and scale bit field for the active slots, this may or may not result in writing a result to the FIFO. When the trigger source is an external pin then one external pin transition of the proper polarity will result in one complete scan of all enabled slots. If the external pin is connected to a repetitive pulse source then repeating scans of all enabled slots are run at the input trigger rate.

### 12.4.2 Repeat Mode

Counter/Timer 3A has a special bit in its configuration register that allows it to be a source of repetitive triggers for the ADC. If counter/timer 3 is initialized for this purpose then one only needs to turn on the RPTEN bit in the ADC configuration registers to enable this mode in the ADC.

**NOTE**

The mode controller does **not** process these repetitive triggers from the counter/ timer until a first triggering event occurs from the normal trigger sources. Thus one can select software triggering in the TRIGSEL field and set up all of the other ADC registers for the desired sample acquisitions. Then one can write to the software trigger register and the mode controller will enter REPEAT mode. In repeat mode, the mode controller waits only for each successive counter/timer 3A input to launch a scan of all enabled slots.

### 12.4.3 Low Power Modes

While the OPMODE and LPMODE bit fields are fed directly to the ADC analog side, the LPMODE field also affects how the mode controller handles power down cycles between conversions. For example, setting LPMODE to zero causes the mode controller to keep the ADC powered up between conversions.

Setting the LPMODE field to one causes the mode controller to do a partial shutdown of the block. This mode requires approximately 50 micro-seconds to revive the ADC and thus adds to the latency from the trigger event. Setting the LPMODE to a value of two disconnects the clocks and all power from the ADC effectively eliminating all active and leakage current consumption.

## 12.5 Interrupts

The ADC has 6 interrupt status bits with corresponding interrupt enable bits, as follows:

1. Conversion Complete Interrupt
2. Scan Complete Interrupt
3. FIFO Overflow Level 1
4. FIFO Overflow Level 2
5. Window Comparator Excursion Interrupt (a.k.a. outside interrupt)
6. Window Comparator Incursion Interrupt (a.k.a. inside interrupt)

The window comparator interrupts are discussed above. See “Window Comparator” on page 318.

There are two interrupts based on the *fullness* of the FIFO. When the respective interrupts are enabled, Overflow 1 fires when the FIFO reaches 75% full, viz. 6 entries. Overflow 2 fires when the FIFO is completely full.

When enabled, the conversion complete interrupt fires when a single slot completes its conversion and the resulting conversion data is pushed into the FIFO.

When enabled, the scan complete interrupt indicates that all enabled slots have sampled their respective channels following a trigger event.

When a single slot is enabled and programmed to average over exactly one measurement and the scan complete and conversion complete interrupts are enabled, a trigger event will result in the conversion complete and scan complete interrupts firing simultaneously upon completion of the ADC scan. Again, if both respective interrupts are enabled and a single slot is enabled and programmed to average over 128 measurements, 128 trigger events result in 128 scan complete interrupts and exactly one conversion complete interrupt following the 128 ADC scans. When multiple slots are enabled with different settings for the number of measurements to average, the conversion complete interrupt signifies that one or more of the conversions have completed and the FIFO contains valid data for one or more of the slot conversions.

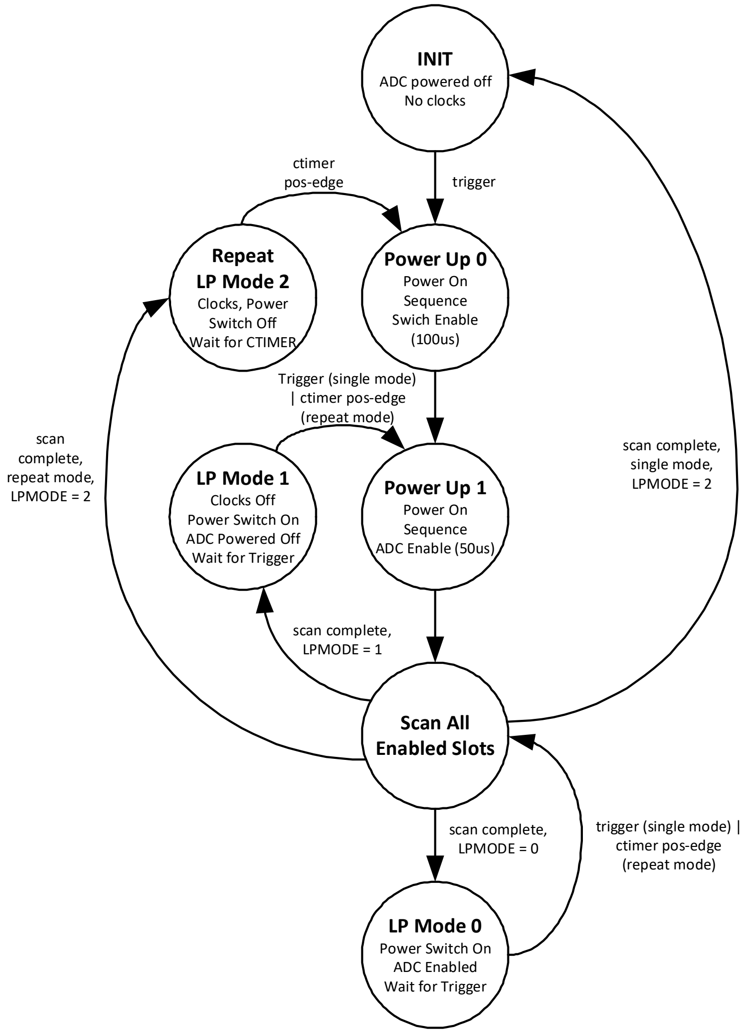
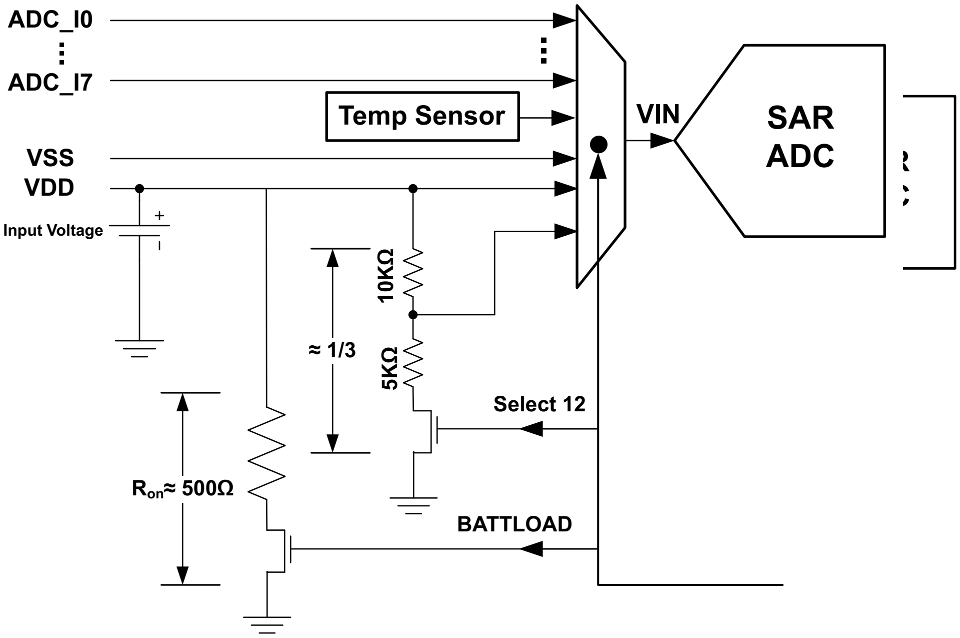


**Figure 51. ADC State Diagram**

## 12.6 Voltage Divider and Switchable Battery Load

The Apollo SoC's ADC includes a switchable voltage divider that enables the ADC to measure the input voltage to the VDD rail. In most systems this will be the battery voltage applied to the SoC. The voltage divider is only switched on when one of the active slots is selecting analog mux channel 12. That is only when the mode controller is ultimately triggered and powers up the ADC block for a conversion scan of all active slots. Otherwise, the voltage divider is turned off.

**Figure 52. Switchable Battery Load**



The switchable load resistor is enabled by the BATTLOAD bit described in Table 501, “CFG Register Bits,” on page 325.

This feature is used to help estimate the health of the battery chemistry by estimating the internal resistance of the battery.

## 12.7 ADC Registers

### Analog Digital Converter Control

**INSTANCE 0 BASE ADDRESS:**0x50008000

This is the detailed description of the Analog Digital Converter Register Block. The ADC Register Block contains the software control for enablement, slot configuration, clock configuration, trigger configuration, temperature sensor enablement, power modes, accumulate/divide, window comparison and interrupt control for the ADC functional unit.

### 12.7.1 Register Memory Map

**Table 499: ADC Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0x50008000 | CFG | Configuration Register |
| 0x50008004 | STAT | ADC Power Status |
| 0x50008008 | SWT | Software trigger |
| 0x5000800C | SL0CFG | Slot 0 Configuration Register |
| 0x50008010 | SL1CFG | Slot 1 Configuration Register |
| 0x50008014 | SL2CFG | Slot 2 Configuration Register |
| 0x50008018 | SL3CFG | Slot 3 Configuration Register |
| 0x5000801C | SL4CFG | Slot 4 Configuration Register |
| 0x50008020 | SL5CFG | Slot 5 Configuration Register |
| 0x50008024 | SL6CFG | Slot 6 Configuration Register |
| 0x50008028 | SL7CFG | Slot 7 Configuration Register |
| 0x5000802C | WLIM | Window Comparator Limits Register |
| 0x50008030 | FIFO | FIFO Data and Valid Count Register |
| 0x50008200 | INTEN | ADC Interrupt registers: Enable |
| 0x50008204 | INTSTAT | ADC Interrupt registers: Status |
| 0x50008208 | INTCLR | ADC Interrupt registers: Clear |
| 0x5000820C | INTSET | ADC Interrupt registers: Set |

### 12.7.2 ADC Registers

#### 12.7.2.1 CFG Register

**Configuration Register**

**OFFSET:** 0x00000000

**INSTANCE 0 ADDRESS:** 0x50008000

The ADC Configuration Register contains the software control for selecting the clock frequency used for the SAR conversions, the trigger polarity, the trigger select, the reference voltage select, the operating mode (current control for high frequency conversions), the low power mode, the operating mode (single scan per trigger vs. repeating mode), temperature sensor enable and ADC enable.

**Table 500: CFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | CLKSEL | | | RSVD | | | TRIGPOL | TRIGSEL | | | | RSVD | | | | | | REFSEL | | BATTLOAD | OPMODE | | LPMODE | | RPTEN | TMPSPWR | ADCEN |

### Table 501: CFG Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:27 | RSVD | 0x0 | RO | RESERVED. |
| 26:24 | CLKSEL | 0x0 | RW | Select the source and frequency for the ADC clock. All values not enumerated below are undefined.<br>OFF = 0x0 - Low Power Mode.<br>12MHZ = 0x1 - 12 MHz ADC clock.<br>6MHZ = 0x2 - 6 MHz ADC clock.<br>3MHZ = 0x3 - 12 MHz ADC clock.<br>1_5MHZ = 0x4 - 1.5 MHz ADC clock. |
| 23:21 | RSVD | 0x0 | RO | RESERVED. |
| 20 | TRIGPOL | 0x0 | RW | This bit selects the ADC trigger polarity for external off chip triggers.<br>RISING_EDGE = 0x0 - Trigger on rising edge.<br>FALLING_EDGE = 0x1 - Trigger on falling edge. |
| 19:16 | TRIGSEL | 0x0 | RW | Select the ADC trigger source.<br>EXT0 = 0x0 - Off chip External Trigger0 (ADC_ET0)<br>EXT1 = 0x1 - Off chip External Trigger1 (ADC_ET1)<br>EXT2 = 0x2 - Off chip External Trigger2 (ADC_ET2)<br>EXT3 = 0x3 - Off chip External Trigger3 (ADC_ET3)<br>EXT4 = 0x4 - Off chip External Trigger4 (ADC_ET4)<br>EXT5 = 0x5 - Off chip External Trigger5 (ADC_ET5)<br>EXT6 = 0x6 - Off chip External Trigger6 (ADC_ET6)<br>EXT7 = 0x7 - Off chip External Trigger7 (ADC_ET7)<br>SWT = 0x8 - Software Trigger |
| 15:10 | RSVD | 0x0 | RO | RESERVED. |
| 9:8 | REFSEL | 0x0 | RW | Select the ADC reference voltage.<br>INTERNAL = 0x0 - Internal Bandgap Reference Voltage<br>VDD = 0x1 - Select VDD as the ADEC reference voltage.<br>ADCREF = 0x2 - Off Chip Reference (ADC_REF)<br>UNDEFINED = 0x3 - Reserved |
| 7 | BATTLOAD | 0x0 | RW | Control 400 Ohm battery load resistor.<br>EN = 0x0 - Enable battery load.<br>DIS = 0x1 - Disable battery load. |
| 6:5 | OPMODE | 0x0 | RW | Select the sample rate mode. It adjusts the current in the ADC for higher sample rates. A 12 MHz ADC clock can result in a sample rate up to 1Msps depending on the trigger or repeating mode rate. A 1.5 MHz ADC clock can result in a sample rate up 125 KSPS.<br>NOTE: All other possible values not specified below, including the default value, are undefined or unsupported. This field must be set to 0x2 for proper operation.<br>SAMPLE_RATE_125K_1MSPS = 0x2 - Sample Rate 125 KSPS to 1 MSPS |

**Table 501: CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 4:3 | LPMODE | 0x0 | RW | Select power mode to enter between active scans.<br><br>MODE0 = 0x0 - Low Power Mode 0 (2'b00). Leaves the ADC fully powered between scans with no latency between a trigger event and sample data collection.<br>MODE1 = 0x1 - Low Power Mode 1 (2'b01). Enables a low power mode for the ADC between scans requiring 50us initialization time (latency) between a trigger event and the scan (assuming the HFRC remains running and the MCU is not in deepsleep mode in which case additional startup latency for HFRC startup is required).<br>MODE2 = 0x2 - Low Power Mode 2 (2'b10). Disconnects power and clocks to the ADC effectively eliminating all active power associated with the ADC between scans. This mode requires 150us initialization (again, assuming the HFRC remains running and the MCU is not in deepsleep mode in which case additional startup latency for HFRC startup is required).<br>MODE_UNDEFINED = 0x3 - Undefined Mode (2'b11) |
| 2 | RPTEN | 0x0 | RW<br>X | This bit enables Repeating Scan Mode.<br><br>SINGLE_SCAN = 0x0 - In Single Scan Mode, the ADC will complete a single scan upon each trigger event.<br>REPEATING_SCAN = 0x1 - In Repeating Scan Mode, the ADC will complete it's first scan upon the initial trigger event and all subsequent scans will occur at regular intervals defined by the configuration programmed for the CTTMRA3 internal timer until the timer is disabled or the ADC is disabled. |
| 1 | TMPSPWR | 0x0 | RW | This enables power to the temperature sensor module. After setting this bit, the temperature sensor will remain powered down while the ADC is power is disconnected (i.e, when the ADC PWDSTAT is 2'b10).<br><br>DIS = 0x0 - Power down the temperature sensor.<br>EN = 0x1 - Enable the temperature sensor when the ADC is in it's active state. |
| 0 | ADCEN | 0x0 | RW | This bit enables the ADC module. While the ADC is enabled, the ADCCFG and SLOT Configuration regsiter settings must remain stable and unchanged.<br><br>DIS = 0x0 - Disable the ADC module.<br>EN = 0x1 - Enable the ADC module. |

### *12.7.2.2 STAT Register*

**ADC Power Status**

**OFFSET:** 0x00000004

**INSTANCE 0 ADDRESS:** 0x50008004

This register indicates the power status for the ADC. ADC power mode 0 indicates the ADC is in it's full power state and is ready to process scans. ADC Power mode 1 indicated the ADC power switch is on, but the ADC SAR module is in a low power state. From this state, a trigger will bring the ADC into it's active state after a 50us calibration period. ADC power mode 2 indicates the ADC is in it's lowest power state.

**Table 502: STAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | PWDSTAT | |

**Table 503: STAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:2 | RSVD | 0x0 | RO | RESERVED. |
| 1:0 | PWDSTAT | 0x0 | RO | Indicates the power-status of the ADC.<br>ON = 0x0 - Powered on.<br>SWITCH_ON_SAR_OFF = 0x1 - Power switch on, ADC Low Power Mode 1.<br>POWER_SWITCH_OFF = 0x2 - Power switch off, ADC disabled. |

### 12.7.2.3 *SWT Register*

**Software trigger**

**OFFSET:** 0x00000008

**INSTANCE 0 ADDRESS:** 0x50008008

This register enables initiating an ADC scan through software.

**Table 504: SWT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | SWT | | | | | | | |

**Table 505: SWT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:8 | RSVD | 0x0 | RO | RESERVED. |
| 7:0 | SWT | 0x0 | RW | Writing 0x37 to this register generates a software trigger.<br>GEN_SW_TRIGGER = 0x37 - Writing this value generates a software trigger. |

### *12.7.2.4 SL0CFG Register*

**Slot 0 Configuration Register**

**OFFSET:** 0x0000000C

**INSTANCE 0 ADDRESS:** 0x5000800C

Slot 0 Configuration Register

**Table 506: SL0CFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | ADSEL0 | | | RSVD | | | | | THSEL0 | | | RSVD | | | | CHSEL0 | | | | RSVD | | | | | | WCEN0 | SLEN0 |

**Table 507: SL0CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:27 | RSVD | 0x0 | RO | RESERVED. |
| 26:24 | ADSEL0 | 0x0 | RW | Select the number of measurements to average in the accumulate divide module for this slot.<br><br>AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot.<br>AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot.<br>AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot.<br>AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot.<br>AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot.<br>AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot.<br>AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot.<br>AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot. |
| 23:19 | RSVD | 0x0 | RO | RESERVED. |
| 18:16 | THSEL0 | 0x0 | RW | Select the track and hold delay for this slot.NOTE: The track and hold delay must be less than 50us for correct operation. When the ADC is configured to use the 1.5Mhz clock, the track and hold delay cannot exceed 64 clocks.<br><br>1_ADC_CLK = 0x0 - 1 ADC clock cycle.<br>2_ADC_CLKS = 0x1 - 2 ADC clock cycles.<br>4_ADC_CLKS = 0x2 - 4 ADC clock cycles.<br>8_ADC_CLKS = 0x3 - 8 ADC clock cycles.<br>16_ADC_CLKS = 0x4 - 16 ADC clock cycles.<br>32_ADC_CLKS = 0x5 - 32 ADC clock cycles.<br>64_ADC_CLKS = 0x6 - 64 ADC clock cycles.<br>128_ADC_CLKS = 0x7 - 128 ADC clock cycles. |

**Table 507: SL0CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 15:12 | RSVD | 0x0 | RO | RESERVED. |
| 11:8 | CHSEL0 | 0x0 | RW | Select one of the 13 channel inputs for this slot.<br><br>EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection.<br>EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection.<br>EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection.<br>EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection.<br>EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection.<br>EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection.<br>EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection.<br>EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection.<br>TEMP = 0x8 - ADC_TEMP internal temperature sensor.<br>VDD = 0x9 - ADC_VDD internal power rail connection.<br>VSS = 0xA - ADC_VSS internal ground connection.<br>VBATT = 0xC - ADC_VBATT internal voltage divide-by-3 connection to input power rail. |
| 7:2 | RSVD | 0x0 | RO | RESERVED. |
| 1 | WCEN0 | 0x0 | RW | This bit enables the window compare function for slot 0.<br><br>WCEN = 0x1 - Enable the window compare for slot 0. |
| 0 | SLEN0 | 0x0 | RW | This bit enables slot 0 for ADC conversions.<br><br>SLEN = 0x1 - Enable slot 0 for ADC conversions. |

### *12.7.2.5 SL1CFG Register*

**Slot 1 Configuration Register**

**OFFSET:** 0x00000010

**INSTANCE 0 ADDRESS:** 0x50008010

Slot 1 Configuration Register

**Table 508: SL1CFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | ADSEL1 | | | RSVD | | | | | THSEL1 | | | RSVD | | | | CHSEL1 | | | | RSVD | | | | | | WCEN1 | SLEN1 |

**Table 509: SL1CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:27 | RSVD | 0x0 | RO | RESERVED. |

**Table 509: SL1CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 26:24 | ADSEL1 | 0x0 | RW | Select the number of measurements to average in the accumulate divide module for this slot.<br><br>AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot.<br>AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot.<br>AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot.<br>AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot.<br>AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot.<br>AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot.<br>AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot.<br>AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot. |
| 23:19 | RSVD | 0x0 | RO | RESERVED. |
| 18:16 | THSEL1 | 0x0 | RW | Select the track and hold delay for this slot.NOTE: The track and hold delay must be less than 50us for correct operation. When the ADC is configured to use the 1.5 Mhz clock, the track and hold delay cannot exceed 64 clocks.<br><br>1_ADC_CLK = 0x0 - 1 ADC clock cycle.<br>2_ADC_CLKS = 0x1 - 2 ADC clock cycles.<br>4_ADC_CLKS = 0x2 - 4 ADC clock cycles.<br>8_ADC_CLKS = 0x3 - 8 ADC clock cycles.<br>16_ADC_CLKS = 0x4 - 16 ADC clock cycles.<br>32_ADC_CLKS = 0x5 - 32 ADC clock cycles.<br>64_ADC_CLKS = 0x6 - 64 ADC clock cycles.<br>128_ADC_CLKS = 0x7 - 128 ADC clock cycles. |
| 15:12 | RSVD | 0x0 | RO | RESERVED. |
| 11:8 | CHSEL1 | 0x0 | RW | Select one of the 13 channel inputs for this slot.<br><br>EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection.<br>EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection.<br>EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection.<br>EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection.<br>EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection.<br>EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection.<br>EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection.<br>EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection.<br>TEMP = 0x8 - ADC_TEMP internal temperature sensor.<br>VDD = 0x9 - ADC_VDD internal power rail connection.<br>VSS = 0xA - ADC_VSS internal ground connection.<br>VBATT = 0xC - ADC_VBATT internal voltage divide-by-3 connection to input power rail. |
| 7:2 | RSVD | 0x0 | RO | RESERVED. |
| 1 | WCEN1 | 0x0 | RW | This bit enables the window compare function for slot 1.<br><br>WCEN = 0x1 - Enable the window compare for slot 1. |

**Table 509: SL1CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 0 | SLEN1 | 0x0 | RW | This bit enables slot 1 for ADC conversions.<br>SLEN = 0x1 - Enable slot 1 for ADC conversions. |

### *12.7.2.6 SL2CFG Register*

**Slot 2 Configuration Register**

**OFFSET:** 0x00000014

**INSTANCE 0 ADDRESS:** 0x50008014

Slot 2 Configuration Register

**Table 510: SL2CFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | ADSEL2 | | | RSVD | | | | | THSEL2 | | | RSVD | | | | CHSEL2 | | | | RSVD | | | | | | WCEN2 | SLEN2 |

**Table 511: SL2CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:27 | RSVD | 0x0 | RO | RESERVED. |
| 26:24 | ADSEL2 | 0x0 | RW | Select the number of measurements to average in the accumulate divide module for this slot.<br>AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot.<br>AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot.<br>AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot.<br>AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot.<br>AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot.<br>AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot.<br>AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot.<br>AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot. |
| 23:19 | RSVD | 0x0 | RO | RESERVED. |

**Table 511: SL2CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 18:16 | THSEL2 | 0x0 | RW | Select the track and hold delay for this slot.NOTE: The track and hold delay must be less than 50us for correct operation. When the ADC is configured to use the 1.5Mhz clock, the track and hold delay cannot exceed 64 clocks.<br><br>1_ADC_CLK = 0x0 - 1 ADC clock cycle.<br>2_ADC_CLKS = 0x1 - 2 ADC clock cycles.<br>4_ADC_CLKS = 0x2 - 4 ADC clock cycles.<br>8_ADC_CLKS = 0x3 - 8 ADC clock cycles.<br>16_ADC_CLKS = 0x4 - 16 ADC clock cycles.<br>32_ADC_CLKS = 0x5 - 32 ADC clock cycles.<br>64_ADC_CLKS = 0x6 - 64 ADC clock cycles.<br>128_ADC_CLKS = 0x7 - 128 ADC clock cycles. |
| 15:12 | RSVD | 0x0 | RO | RESERVED. |
| 11:8 | CHSEL2 | 0x0 | RW | Select one of the 13 channel inputs for this slot.<br><br>EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection.<br>EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection.<br>EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection.<br>EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection.<br>EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection.<br>EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection.<br>EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection.<br>EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection.<br>TEMP = 0x8 - ADC_TEMP internal temperature sensor.<br>VDD = 0x9 - ADC_VDD internal power rail connection.<br>VSS = 0xA - ADC_VSS internal ground connection.<br>VBATT = 0xC - ADC_VBATT internal voltage divide-by-3 connection to input power rail. |
| 7:2 | RSVD | 0x0 | RO | RESERVED. |
| 1 | WCEN2 | 0x0 | RW | This bit enables the window compare function for slot 2.<br><br>WCEN = 0x1 - Enable the window compare for slot 2. |
| 0 | SLEN2 | 0x0 | RW | This bit enables slot 2 for ADC conversions.<br><br>SLEN = 0x1 - Enable slot 2 for ADC conversions. |

### *12.7.2.7 SL3CFG Register*

**Slot 3 Configuration Register**

**OFFSET:** 0x00000018

**INSTANCE 0 ADDRESS:** 0x50008018

Slot 3 Configuration Register

### Table 512: SL3CFG Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | ADSEL3 | | | RSVD | | | | | THSEL3 | | | RSVD | | | | CHSEL3 | | | | RSVD | | | | | | WCEN3 | SLEN3 |

### Table 513: SL3CFG Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:27 | RSVD | 0x0 | RO | RESERVED. |
| 26:24 | ADSEL3 | 0x0 | RW | Select the number of measurements to average in the accumulate divide module for this slot.<br><br>AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot.<br>AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot.<br>AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot.<br>AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot.<br>AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot.<br>AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot.<br>AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot.<br>AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot. |
| 23:19 | RSVD | 0x0 | RO | RESERVED. |
| 18:16 | THSEL3 | 0x0 | RW | Select the track and hold delay for this slot.NOTE: The track and hold delay must be less than 50us for correct operation. When the ADC is configured to use the 1.5Mhz clock, the track and hold delay cannot exceed 64 clocks.<br><br>1_ADC_CLK = 0x0 - 1 ADC clock cycle.<br>2_ADC_CLKS = 0x1 - 2 ADC clock cycles.<br>4_ADC_CLKS = 0x2 - 4 ADC clock cycles.<br>8_ADC_CLKS = 0x3 - 8 ADC clock cycles.<br>16_ADC_CLKS = 0x4 - 16 ADC clock cycles.<br>32_ADC_CLKS = 0x5 - 32 ADC clock cycles.<br>64_ADC_CLKS = 0x6 - 64 ADC clock cycles.<br>128_ADC_CLKS = 0x7 - 128 ADC clock cycles. |
| 15:12 | RSVD | 0x0 | RO | RESERVED. |

**Table 513: SL3CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 11:8 | CHSEL3 | 0x0 | RW | Select one of the 13 channel inputs for this slot.<br><br>EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection.<br>EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection.<br>EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection.<br>EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection.<br>EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection.<br>EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection.<br>EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection.<br>EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection.<br>TEMP = 0x8 - ADC_TEMP internal temperature sensor.<br>VDD = 0x9 - ADC_VDD internal power rail connection.<br>VSS = 0xA - ADC_VSS internal ground connection.<br>VBATT = 0xC - ADC_VBATT internal voltage divide-by-3 connection to input power rail. |
| 7:2 | RSVD | 0x0 | RO | RESERVED. |
| 1 | WCEN3 | 0x0 | RW | This bit enables the window compare function for slot 3.<br><br>WCEN = 0x1 - Enable the window compare for slot 3. |
| 0 | SLEN3 | 0x0 | RW | This bit enables slot 3 for ADC conversions.<br><br>SLEN = 0x1 - Enable slot 3 for ADC conversions. |

### 12.7.2.8 SL4CFG Register

**Slot 4 Configuration Register**

**OFFSET:** 0x0000001C

**INSTANCE 0 ADDRESS:** 0x5000801C

Slot 4 Configuration Register

**Table 514: SL4CFG Register**

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>09</th><th>08</th><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="5">RSVD</td><td colspan="3">ADSEL4</td><td colspan="5">RSVD</td><td colspan="3">THSEL4</td><td colspan="4">RSVD</td><td colspan="4">CHSEL4</td><td colspan="6">RSVD</td><td>WCEN4</td><td>SLEN4</td></tr>
</table>

**Table 515: SL4CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:27 | RSVD | 0x0 | RO | RESERVED. |

**Table 515: SL4CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 26:24 | ADSEL4 | 0x0 | RW | Select the number of measurements to average in the accumulate divide module for this slot.<br><br>AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot.<br>AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot.<br>AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot.<br>AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot.<br>AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot.<br>AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot.<br>AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot.<br>AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot. |
| 23:19 | RSVD | 0x0 | RO | RESERVED. |
| 18:16 | THSEL4 | 0x0 | RW | Select the track and hold delay for this slot.NOTE: The track and hold delay must be less than 50us for correct operation.  When the ADC is configured to use the 1.5Mhz clock, the track and hold delay cannot exceed 64 clocks.<br><br>1_ADC_CLK = 0x0 - 1 ADC clock cycle.<br>2_ADC_CLKS = 0x1 - 2 ADC clock cycles.<br>4_ADC_CLKS = 0x2 - 4 ADC clock cycles.<br>8_ADC_CLKS = 0x3 - 8 ADC clock cycles.<br>16_ADC_CLKS = 0x4 - 16 ADC clock cycles.<br>32_ADC_CLKS = 0x5 - 32 ADC clock cycles.<br>64_ADC_CLKS = 0x6 - 64 ADC clock cycles.<br>128_ADC_CLKS = 0x7 - 128 ADC clock cycles. |
| 15:12 | RSVD | 0x0 | RO | RESERVED. |
| 11:8 | CHSEL4 | 0x0 | RW | Select one of the 13 channel inputs for this slot.<br><br>EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection.<br>EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection.<br>EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection.<br>EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection.<br>EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection.<br>EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection.<br>EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection.<br>EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection.<br>TEMP = 0x8 - ADC_TEMP internal temperature sensor.<br>VDD = 0x9 - ADC_VDD internal power rail connection.<br>VSS = 0xA - ADC_VSS internal ground connection.<br>VBATT = 0xC - ADC_VBATT internal voltage divide-by-3 connection to input power rail. |
| 7:2 | RSVD | 0x0 | RO | RESERVED. |
| 1 | WCEN4 | 0x0 | RW | This bit enables the window compare function for slot 4.<br><br>WCEN = 0x1 - Enable the window compare for slot 4. |

**Table 515: SL4CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 0 | SLEN4 | 0x0 | RW | This bit enables slot 4 for ADC conversions.<br>SLEN = 0x1 - Enable slot 4 for ADC conversions. |

#### *12.7.2.9 SL5CFG Register*

**Slot 5 Configuration Register**

**OFFSET:** 0x00000020

**INSTANCE 0 ADDRESS:** 0x50008020

Slot 5 Configuration Register

**Table 516: SL5CFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | ADSEL5 | | | RSVD | | | | | THSEL5 | | | RSVD | | | | CHSEL5 | | | | RSVD | | | | | | WCEN5 | SLEN5 |

**Table 517: SL5CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:27 | RSVD | 0x0 | RO | RESERVED. |
| 26:24 | ADSEL5 | 0x0 | RW | Select number of measurements to average in the accumulate divide module for this slot.<br>AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot.<br>AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot.<br>AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot.<br>AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot.<br>AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot.<br>AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot.<br>AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot.<br>AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot. |
| 23:19 | RSVD | 0x0 | RO | RESERVED. |

Table 517: SL5CFG Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 18:16 | THSEL5 | 0x0 | RW | Select track and hold delay for this slot.NOTE: The track and hold delay must be less than 50us for correct operation. When the ADC is configured to use the 1.5Mhz clock, the track and hold delay cannot exceed 64 clocks.<br>1_ADC_CLK = 0x0 - 1 ADC clock cycle.<br>2_ADC_CLKS = 0x1 - 2 ADC clock cycles.<br>4_ADC_CLKS = 0x2 - 4 ADC clock cycles.<br>8_ADC_CLKS = 0x3 - 8 ADC clock cycles.<br>16_ADC_CLKS = 0x4 - 16 ADC clock cycles.<br>32_ADC_CLKS = 0x5 - 32 ADC clock cycles.<br>64_ADC_CLKS = 0x6 - 64 ADC clock cycles.<br>128_ADC_CLKS = 0x7 - 128 ADC clock cycles. |
| 15:12 | RSVD | 0x0 | RO | RESERVED. |
| 11:8 | CHSEL5 | 0x0 | RW | Select one of the 13 channel inputs for this slot.<br>EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection.<br>EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection.<br>EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection.<br>EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection.<br>EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection.<br>EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection.<br>EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection.<br>EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection.<br>TEMP = 0x8 - ADC_TEMP internal temperature sensor.<br>VDD = 0x9 - ADC_VDD internal power rail connection.<br>VSS = 0xA - ADC_VSS internal ground connection.<br>VBATT = 0xC - ADC_VBATT internal voltage divide-by-3 connection to input power rail. |
| 7:2 | RSVD | 0x0 | RO | RESERVED. |
| 1 | WCEN5 | 0x0 | RW | This bit enables the window compare function for slot 5.<br>WCEN = 0x1 - Enable the window compare for slot 5. |
| 0 | SLEN5 | 0x0 | RW | This bit enables slot 5 for ADC conversions.<br>SLEN = 0x1 - Enable slot 5 for ADC conversions. |

### *12.7.2.10SL6CFG Register*

**Slot 6 Configuration Register**

**OFFSET:** 0x00000024

**INSTANCE 0 ADDRESS:** 0x50008024

Slot 6 Configuration Register

### Table 518: SL6CFG Register

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | ADSEL6 | | | RSVD | | | | | THSEL6 | | | RSVD | | | | CHSEL6 | | | | RSVD | | | | | | WCEN6 | SLEN6 |

### Table 519: SL6CFG Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:27 | RSVD | 0x0 | RO | RESERVED. |
| 26:24 | ADSEL6 | 0x0 | RW | Select the number of measurements to average in the accumulate divide module for this slot.<br>AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot.<br>AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot.<br>AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot.<br>AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot.<br>AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot.<br>AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot.<br>AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot.<br>AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot. |
| 23:19 | RSVD | 0x0 | RO | RESERVED. |
| 18:16 | THSEL6 | 0x0 | RW | Select track and hold delay for this slot.NOTE: The track and hold delay must be less than 50us for correct operation.  When the ADC is configured to use the 1.5Mhz clock, the track and hold delay cannot exceed 64 clocks.<br>1_ADC_CLK = 0x0 - 1 ADC clock cycle.<br>2_ADC_CLKS = 0x1 - 2 ADC clock cycles.<br>4_ADC_CLKS = 0x2 - 4 ADC clock cycles.<br>8_ADC_CLKS = 0x3 - 8 ADC clock cycles.<br>16_ADC_CLKS = 0x4 - 16 ADC clock cycles.<br>32_ADC_CLKS = 0x5 - 32 ADC clock cycles.<br>64_ADC_CLKS = 0x6 - 64 ADC clock cycles.<br>128_ADC_CLKS = 0x7 - 128 ADC clock cycles. |
| 15:12 | RSVD | 0x0 | RO | RESERVED. |

**Table 519: SL6CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 11:8 | CHSEL6 | 0x0 | RW | Select one of the 13 channel inputs for this slot.<br><br>EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection.<br>EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection.<br>EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection.<br>EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection.<br>EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection.<br>EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection.<br>EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection.<br>EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection.<br>TEMP = 0x8 - ADC_TEMP internal temperature sensor.<br>VDD = 0x9 - ADC_VDD internal power rail connection.<br>VSS = 0xA - ADC_VSS internal ground connection.<br>VBATT = 0xC - ADC_VBATT internal voltage divide-by-3 connection to input power rail. |
| 7:2 | RSVD | 0x0 | RO | RESERVED. |
| 1 | WCEN6 | 0x0 | RW | This bit enables the window compare function for slot 6.<br><br>WCEN = 0x1 - Enable the window compare for slot 6. |
| 0 | SLEN6 | 0x0 | RW | This bit enables slot 6 for ADC conversions.<br><br>SLEN = 0x1 - Enable slot 6 for ADC conversions. |

### *12.7.2.11SL7CFG Register*

**Slot 7 Configuration Register**

**OFFSET:** 0x00000028

**INSTANCE 0 ADDRESS:** 0x50008028

Slot 7 Configuration Register

**Table 520: SL7CFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | ADSEL7 | | | RSVD | | | | | THSEL7 | | | RSVD | | | | CHSEL7 | | | | RSVD | | | | | | WCEN7 | SLEN7 |

**Table 521: SL7CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:27 | RSVD | 0x0 | RO | RESERVED. |

### Table 521: SL7CFG Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 26:24 | ADSEL7 | 0x0 | RW | Select the number of measurements to average in the accumulate divide module for this slot.<br><br>AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot.<br>AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot.<br>AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot.<br>AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot.<br>AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot.<br>AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot.<br>AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot.<br>AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot. |
| 23:19 | RSVD | 0x0 | RO | RESERVED. |
| 18:16 | THSEL7 | 0x0 | RW | Select track and hold delay for this slot.NOTE: The track and hold delay must be less than 50us for correct operation. When the ADC is configured to use the 1.5Mhz clock, the track and hold delay cannot exceed 64 clocks.<br><br>1_ADC_CLK = 0x0 - 1 ADC clock cycle.<br>2_ADC_CLKS = 0x1 - 2 ADC clock cycles.<br>4_ADC_CLKS = 0x2 - 4 ADC clock cycles.<br>8_ADC_CLKS = 0x3 - 8 ADC clock cycles.<br>16_ADC_CLKS = 0x4 - 16 ADC clock cycles.<br>32_ADC_CLKS = 0x5 - 32 ADC clock cycles.<br>64_ADC_CLKS = 0x6 - 64 ADC clock cycles.<br>128_ADC_CLKS = 0x7 - 128 ADC clock cycles. |
| 15:12 | RSVD | 0x0 | RO | RESERVED. |
| 11:8 | CHSEL7 | 0x0 | RW | Select one of the 13 channel inputs for this slot.<br><br>EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection.<br>EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection.<br>EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection.<br>EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection.<br>EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection.<br>EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection.<br>EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection.<br>EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection.<br>TEMP = 0x8 - ADC_TEMP internal temperature sensor.<br>VDD = 0x9 - ADC_VDD internal power rail connection.<br>VSS = 0xA - ADC_VSS internal ground connection.<br>VBATT = 0xC - ADC_VBATT internal voltage divide-by-3 connection to input power rail. |
| 7:2 | RSVD | 0x0 | RO | RESERVED. |
| 1 | WCEN7 | 0x0 | RW | This bit enables the window compare function for slot 7.<br><br>WCEN = 0x1 - Enable the window compare for slot 7. |

**Table 521: SL7CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 0 | SLEN7 | 0x0 | RW | This bit enables slot 7 for ADC conversions.<br>SLEN = 0x1 - Enable slot 7 for ADC conversions. |

### *12.7.2.12WLIM Register*

**Window Comparator Limits Register**

**OFFSET:** 0x0000002C

**INSTANCE 0 ADDRESS:** 0x5000802C

Window Comparator Limits Register

**Table 522: WLIM Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ULIM | | | | | | | | | | | | | | | | LLIM | | | | | | | | | | | | | | | |

**Table 523: WLIM Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:16 | ULIM | 0x0 | RW | Sets the upper limit for the wondow comparator. |
| 15:0 | LLIM | 0x0 | RW | Sets the lower limit for the wondow comparator. |

### *12.7.2.13FIFO Register*

**FIFO Data and Valid Count Register**

**OFFSET:** 0x00000030

**INSTANCE 0 ADDRESS:** 0x50008030

The ADC FIFO Register contains the slot number and fifo data for the oldest conversion data in the FIFO. The COUNT field indicates the total number of valid entries in the FIFO. A write to this register will pop one of the FIFO entries off the FIFO and decrease the COUNT by 1 if the COUNT is greater than zero.

**Table 524: FIFO Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD_27 | | | | | SLOTNUM | | | RSVD_20 | | | | COUNT | | | | DATA | | | | | | | | | | | | | | | |

**Table 525: FIFO Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:27 | RSVD_27 | 0x0 | RO | RESERVED. |
| 26:24 | SLOTNUM | 0x0 | RO | Slot number associated with this FIFO data. |
| 23:20 | RSVD_20 | 0x0 | RO | RESERVED. |
| 19:16 | COUNT | 0x0 | RO | Number of valid entries in the ADC FIFO. |
| 15:0 | DATA | 0x0 | RO | Oldest data in the FIFO. |

### *12.7.2.14INTEN Register*

**ADC Interrupt registers: Enable**

**OFFSET:** 0x00000200

**INSTANCE 0 ADDRESS:** 0x50008200

Set bits in this register to allow this module to generate the corresponding interrupt.

**Table 526: INTEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | WCINC | WCEXC | FIFOOVR2 | FIFOOVR1 | SCNCMP | CNVCMP |

**Table 527: INTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:6 | RSVD | 0x0 | RO | RESERVED. |
| 5 | WCINC | 0x0 | RW | Window comparator voltage incursion interrupt.<br>WCINCINT = 0x1 - Window comparitor voltage incursion interrupt. |
| 4 | WCEXC | 0x0 | RW | Window comparator voltage excursion interrupt.<br>WCEXCINT = 0x1 - Window comparitor voltage excursion interrupt. |
| 3 | FIFOOVR2 | 0x0 | RW | FIFO 100% full interrupt.<br>FIFOFULLINT = 0x1 - FIFO 100% full interrupt. |

**Table 527: INTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 2 | FIFOOVR1 | 0x0 | RW | FIFO 75% full interrupt.<br>FIFO75INT = 0x1 - FIFO 75% full interrupt. |
| 1 | SCNCMP | 0x0 | RW | ADC scan complete interrupt.<br>SCNCMPINT = 0x1 - ADC scan complete interrupt. |
| 0 | CNVCMP | 0x0 | RW | ADC conversion complete interrupt.<br>CNVCMPINT = 0x1 - ADC conversion complete interrupt. |

### *12.7.2.15INTSTAT Register*

**ADC Interrupt registers: Status**

**OFFSET:** 0x00000204

**INSTANCE 0 ADDRESS:** 0x50008204

Read bits from this register to discover the cause of a recent interrupt.

**Table 528: INTSTAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | WCINC | WCEXC | FIFOOVR2 | FIFOOVR1 | SCNCMP | CNVCMP |

**Table 529: INTSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:6 | RSVD | 0x0 | RO | RESERVED. |
| 5 | WCINC | 0x0 | RW | Window comparator voltage incursion interrupt.<br>WCINCINT = 0x1 - Window comparitor voltage incursion interrupt. |
| 4 | WCEXC | 0x0 | RW | Window comparator voltage excursion interrupt.<br>WCEXCINT = 0x1 - Window comparitor voltage excursion interrupt. |
| 3 | FIFOOVR2 | 0x0 | RW | FIFO 100% full interrupt.<br>FIFOFULLINT = 0x1 - FIFO 100% full interrupt. |
| 2 | FIFOOVR1 | 0x0 | RW | FIFO 75% full interrupt.<br>FIFO75INT = 0x1 - FIFO 75% full interrupt. |
| 1 | SCNCMP | 0x0 | RW | ADC scan complete interrupt.<br>SCNCMPINT = 0x1 - ADC scan complete interrupt. |

**Table 529: INTSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 0 | CNVCMP | 0x0 | RW | ADC conversion complete interrupt.<br>CNVCMPINT = 0x1 - ADC conversion complete interrupt. |

### *12.7.2.16INTCLR Register*

**ADC Interrupt registers: Clear**

**OFFSET:** 0x00000208

**INSTANCE 0 ADDRESS:** 0x50008208

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

**Table 530: INTCLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | WCINC | WCEXC | FIFOOVR2 | FIFOOVR1 | SCNCMP | CNVCMP |

**Table 531: INTCLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:6 | RSVD | 0x0 | RO | RESERVED. |
| 5 | WCINC | 0x0 | RW | Window comparator voltage incursion interrupt.<br>WCINCINT = 0x1 - Window comparitor voltage incursion interrupt. |
| 4 | WCEXC | 0x0 | RW | Window comparator voltage excursion interrupt.<br>WCEXCINT = 0x1 - Window comparitor voltage excursion interrupt. |
| 3 | FIFOOVR2 | 0x0 | RW | FIFO 100% full interrupt.<br>FIFOFULLINT = 0x1 - FIFO 100% full interrupt. |
| 2 | FIFOOVR1 | 0x0 | RW | FIFO 75% full interrupt.<br>FIFO75INT = 0x1 - FIFO 75% full interrupt. |
| 1 | SCNCMP | 0x0 | RW | ADC scan complete interrupt.<br>SCNCMPINT = 0x1 - ADC scan complete interrupt. |
| 0 | CNVCMP | 0x0 | RW | ADC conversion complete interrupt.<br>CNVCMPINT = 0x1 - ADC conversion complete interrupt. |

### *12.7.2.17INTSET Register*

**ADC Interrupt registers: Set**

**OFFSET:** 0x0000020C

**INSTANCE 0 ADDRESS:** 0x5000820C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

**Table 532: INTSET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | WCINC | WCEXC | FIFOOVR2 | FIFOOVR1 | SCNCMP | CNVCMP |

**Table 533: INTSET Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:6 | RSVD | 0x0 | RO | RESERVED. |
| 5 | WCINC | 0x0 | RW | Window comparator voltage incursion interrupt.<br>WCINCINT = 0x1 - Window comparitor voltage incursion interrupt. |
| 4 | WCEXC | 0x0 | RW | Window comparator voltage excursion interrupt.<br>WCEXCINT = 0x1 - Window comparitor voltage excursion interrupt. |
| 3 | FIFOOVR2 | 0x0 | RW | FIFO 100% full interrupt.<br>FIFOFULLINT = 0x1 - FIFO 100% full interrupt. |
| 2 | FIFOOVR1 | 0x0 | RW | FIFO 75% full interrupt.<br>FIFO75INT = 0x1 - FIFO 75% full interrupt. |
| 1 | SCNCMP | 0x0 | RW | ADC scan complete interrupt.<br>SCNCMPINT = 0x1 - ADC scan complete interrupt. |
| 0 | CNVCMP | 0x0 | RW | ADC conversion complete interrupt.<br>CNVCMPINT = 0x1 - ADC conversion complete interrupt. |

# 13. Voltage Comparator Module

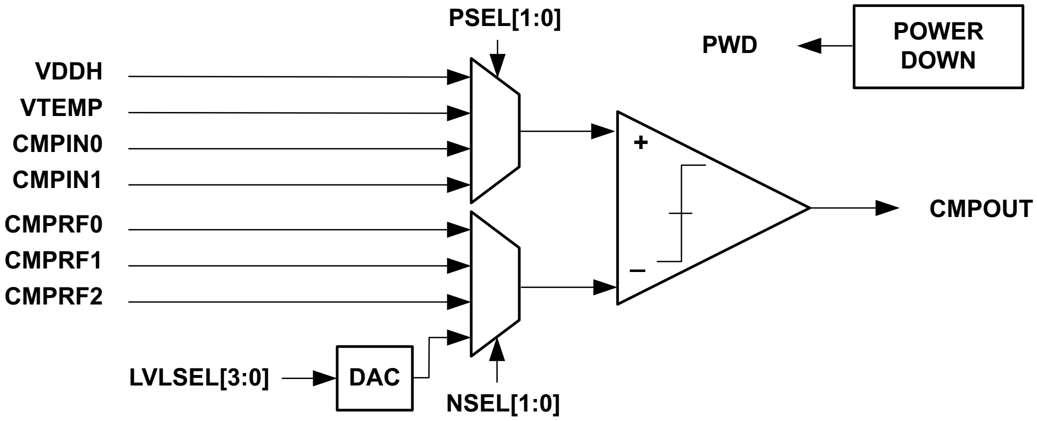


**Figure 53. Block diagram for the Voltage Comparator Module**

## 13.1 Functional Overview

The Voltage Comparator Module, shown in Figure 53, measures a user-selectable voltage at all times. It provides interrupt and software access to the comparator output with multiple options for voltage levels it is comparing. It can be configured to generate an interrupt when the monitored voltage rises above a user-configurable threshold or when the monitored voltage drops below a user-configurable threshold.

The voltage to be monitored is selected by programming the comparator's positive terminal signal, PSEL[1:0], and may be any of:

1. The supply voltage (VDDH)
2. The PTAT voltage from the temperature sensor (VTEMP)
3. External voltage channel 1 (CMPIN0)
4. External voltage channel 2 (CMPIN1)

The reference voltage is selected by programming the comparator's negative terminal, NSEL[1:0] and may be any of:

1. Three external voltage channels (CMPRF0, CMPRF1 or CMPRF2)
2. The internally generated reference (DAC output selected by LVLSEL)

The internal reference voltage is tuned using an on-chip DAC with level select signal LVLSEL[3:0]. When using external inputs or reference inputs, the associated pads must be configured using the GPIO function selects explained in the GPIO document section.

The Voltage Comparator CMPOUT output will remain high while the voltage at the positive input is above the voltage at reference input. The CMPOUT output will transition low when the voltage at the positive input to the comparator falls below the reference input taking into account hysteresis. The CMPOUT output is directly accessible by software by reading the CMPOUT field in the status register. The OUTHI interrupt will be set if enabled and the CMPOUT transitions high or if it is high at the time the interrupt is enabled. Similarly, the OUTLOW interrupt will be set if enabled and the CMPOUT output transitions low or if it is low at the time the interrupt is enabled.

The Voltage Comparator Module is enabled by default and may be powered off by writing 0x37 to the PWDKEY register.

## 13.2 VCOMP Registers

**Voltage Comparator**

**INSTANCE 0 BASE ADDRESS:**0x4000C000

This is the detailed description of the Voltage Comparator Register Block. The Voltage Comparator Register Block contains the software control for selecting the comparator inputs, powerdown control, observing comparator output status and enabling interrupts.

### *13.2.1 Register Memory Map*

**Table 534: VCOMP Register Map**

| Address(s) | Register Name | Description |
|---|---|---|
| 0x4000C000 | CFG | Configuration Register |
| 0x4000C004 | STAT | Status Register |
| 0x4000C008 | PWDKEY | Key Register for Powering Down the Voltage Comparator |
| 0x4000C200 | INTEN | Voltage Comparator Interrupt registers: Enable |
| 0x4000C204 | INTSTAT | Voltage Comparator Interrupt registers: Status |
| 0x4000C208 | INTCLR | Voltage Comparator Interrupt registers: Clear |
| 0x4000C20C | INTSET | Voltage Comparator Interrupt registers: Set |

### 13.2.2 VCOMP Registers

### 13.2.2.1 CFG Register

**Configuration Register**

**OFFSET:** 0x00000000

**INSTANCE 0 ADDRESS:** 0x4000C000

The Voltage Comparator Configuration Register contains the software control for selecting between the four options for the positive input as well as the multiple options for the reference input.

**Table 535: CFG Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | LVLSEL | | | | RSVD | | | | | | NSEL | | RSVD | | | | | | PSEL | |

**Table 536: CFG Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:20 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 19:16 | LVLSEL | 0x0 | RW | When the reference input NSEL is set to NSEL_DAC, this bitfield selects the voltage level for the negative input to the comparator.<br><br>0P58V = 0x0 - Set Reference input to 0.58 Volts.<br>0P77V = 0x1 - Set Reference input to 0.77 Volts.<br>0P97V = 0x2 - Set Reference input to 0.97 Volts.<br>1P16V = 0x3 - Set Reference input to 1.16 Volts.<br>1P35V = 0x4 - Set Reference input to 1.35 Volts.<br>1P55V = 0x5 - Set Reference input to 1.55 Volts.<br>1P74V = 0x6 - Set Reference input to 1.74 Volts.<br>1P93V = 0x7 - Set Reference input to 1.93 Volts.<br>2P13V = 0x8 - Set Reference input to 2.13 Volts.<br>2P32V = 0x9 - Set Reference input to 2.32 Volts.<br>2P51V = 0xA - Set Reference input to 2.51 Volts.<br>2P71V = 0xB - Set Reference input to 2.71 Volts.<br>2P90V = 0xC - Set Reference input to 2.90 Volts.<br>3P09V = 0xD - Set Reference input to 3.09 Volts.<br>3P29V = 0xE - Set Reference input to 3.29 Volts.<br>3P48V = 0xF - Set Reference input to 3.48 Volts. |
| 15:10 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 9:8 | NSEL | 0x0 | RW | This bitfield selects the negative input to the comparator.<br><br>CMPRF0 = 0x0 - Use comparator reference 0 for reference input.<br>CMPRF1 = 0x1 - Use comparator reference 1 for reference input.<br>CMPRF2 = 0x3 - Use comparator reference 2 for reference input.<br>DAC = 0x3 - Use DAC output selected by LVLSEL for reference input. |
| 7:2 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |

Table 536: CFG Register Bits

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 1:0 | PSEL | 0x0 | RW | This bitfield selects the positive input to the comparator.<br><br>VDDADJ = 0x0 - Use VDDADJ for the positive input.<br>VTEMP = 0x1 - Use the temperature sensor output for the positive input.<br>VEXT1 = 0x2 - Use external voltage 1 for positive input.<br>VEXT2 = 0x3 - Use external voltage 1 for positive input. |

### 13.2.2.2 *STAT Register*

**Status Register**

**OFFSET:** 0x00000004

**INSTANCE 0 ADDRESS:** 0x4000C004

Status Register

**Table 537: STAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | PWDSTAT | CMPOUT |

**Table 538: STAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:2 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 1 | PWDSTAT | 0x0 | RO | This bit indicates the power down state of the voltage comparator.<br>POWERED_DOWN = 0x1 - The voltage comparator is powered down. |
| 0 | CMPOUT | 0x0 | RO | This bit is 1 if the positive input of the comparator is greater than the negative input.<br>VOUT_LOW = 0x0 - The negative input of the comparator is greater than the positive input.<br>VOUT_HIGH = 0x1 - The positive input of the comparator is greater than the negative input. |

### 13.2.2.3 *PWDKEY Register*

**Key Register for Powering Down the Voltage Comparator**

**OFFSET:** 0x00000008

**INSTANCE 0 ADDRESS:** 0x4000C008

Key Register for Powering Down the Voltage Comparator

**Table 539: PWDKEY Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PWDKEY | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Table 540: PWDKEY Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:0 | PWDKEY | 0x0 | RW | Key register value.<br>Key = 0x37 - Key |

### *13.2.2.4 INTEN Register*

**Voltage Comparator Interrupt registers: Enable**

**OFFSET:** 0x00000200

**INSTANCE 0 ADDRESS:** 0x4000C200

Set bits in this register to allow this module to generate the corresponding interrupt.

**Table 541: INTEN Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | OUTHI | OUTLOW |

**Table 542: INTEN Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:2 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 1 | OUTHI | 0x0 | RW | This bit is the vcompout high interrupt. |
| 0 | OUTLOW | 0x0 | RW | This bit is the vcompout low interrupt. |

### *13.2.2.5 INTSTAT Register*

**Voltage Comparator Interrupt registers: Status**

**OFFSET:** 0x00000204

**INSTANCE 0 ADDRESS:** 0x4000C204

Read bits from this register to discover the cause of a recent interrupt.

**Table 543: INTSTAT Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | OUTHI | OUTLOW |

**Table 544: INTSTAT Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:2 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 1 | OUTHI | 0x0 | RW | This bit is the vcompout high interrupt. |
| 0 | OUTLOW | 0x0 | RW | This bit is the vcompout low interrupt. |

### *13.2.2.6 INTCLR Register*

**Voltage Comparator Interrupt registers: Clear**

**OFFSET:** 0x00000208

**INSTANCE 0 ADDRESS:** 0x4000C208

Write a 1 to a bit in this register to clear the interrupt status associated with that bit.

**Table 545: INTCLR Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | OUTHI | OUTLOW |

**Table 546: INTCLR Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:2 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 1 | OUTHI | 0x0 | RW | This bit is the vcompout high interrupt. |
| 0 | OUTLOW | 0x0 | RW | This bit is the vcompout low interrupt. |

### *13.2.2.7 INTSET Register*

**Voltage Comparator Interrupt registers: Set**

**OFFSET:** 0x0000020C

**INSTANCE 0 ADDRESS:** 0x4000C20C

Write a 1 to a bit in this register to instantly generate an interrupt from this module. (Generally used for testing purposes).

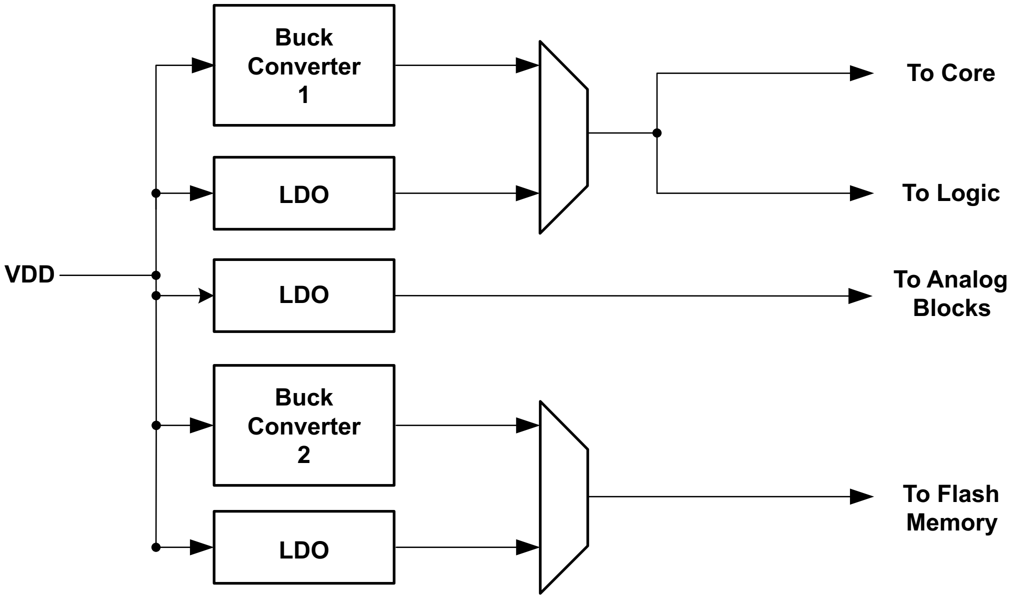
**Table 547: INTSET Register**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSVD | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | OUTHI | OUTLOW |

**Table 548: INTSET Register Bits**

| Bit | Name | Reset | RW | Description |
|---|---|---|---|---|
| 31:2 | RSVD | 0x0 | RO | This bitfield is reserved for future use. |
| 1 | OUTHI | 0x0 | RW | This bit is the vcompout high interrupt. |
| 0 | OUTLOW | 0x0 | RW | This bit is the vcompout low interrupt. |

# 14. Voltage Regulator Module



## 14.1 Functional Overview

The Voltage Regulator Module down-converts and regulates the supply voltage, VDD, with extremely high efficiency. A pair of Buck Converters enables down-conversion from the power supply input (e.g., a battery) at efficiency of >80%. With ultra-low quiescent current, the Buck Converters are optimized for low power environments. The outputs of both Buck Converters are regulated using integrated low dropout linear regulators prior to on-chip power delivery.

The Buck Converters and LDOs of the Voltage Regulator Module are tightly coupled to the various low power modes in the Apollo SoC. When the Apollo SoC enters deep sleep mode, the Buck Converters can be optionally powered down and bypassed, and the LDOs can be placed in an extreme low power mode with only nanoamps of quiescent current.

# 15. Electrical Characteristics

For all tables $T_A$ = -40ºC to 85ºC, Typical values at 25ºC, unless otherwise specified.

## 15.1 Absolute Maximum Ratings

The absolute maximum ratings are the limits to which the device can be subjected without permanently damaging the device and are stress ratings only. Device reliability may be adversely affected by exposure to absolute-maximum ratings for extended periods. Functional operation of the device at the absolute maximum ratings or any other conditions beyond the recommended operating conditions is not implied.

**Table 549: Absolute Maximum Ratings**

| Symbol | Parameter[1] | Test Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| VDDP | Pad supply voltage | | - | 3.8 | V |
| VDDH | Digital supply voltage | | - | 3.8 | V |
| VDDA | Analog supply voltage | | - | 3.8 | V |
| $V_{INPUT}$ | Voltage on any input pin | | 0 | VDDH | V |
| $I_{SRC\_STD}$ | Standard output pin source continuous current | | - | 16 | mA |
| $I_{SINK\_STD}$ | Standard output pin sink continuous current | | - | 16 | mA |
| $I_{HSC\_PWR}$ | High side power switch continuous source current[1] | | - | 50 | mA |
| $I_{HSP\_PWR}$ | High side power switch pulsed source current[1] | 1 ms pulse | | 70 | mA |
| $I_{LSC\_PWR}$ | Low side power switch continuous sink current[2] | | | 50 | mA |
| $I_{LSP\_PWR}$ | Low side power switch pulsed sink current[2] | 1 ms pulse | | 70 | mA |
| $T_S$ | Storage temperature | | -55 | 125 | ºC |
| $T_J$ | Junction temperature | Calculated assuming worst-case power consumption running Coremark, plus two high-side switches passing maximum static current. | -40 | 85.7 | ºC |
| $T_{OP}$ | Operating temperature | | -40 | 85 | ºC |
| $\theta_{JA}$ | Thermal resistance, junction to ambient | BGA Package on 4 layer PCB in still air, 3mW power dissipation | | 76.2 | ºC/W |
| $\theta_{JC}$ | Thermal resistance, junction to package case | BGA Package on 4 layer PCB in still air, 3mW power dissipation | | 17.0 | ºC/W |
| $T_{REFLOW}$ | Reflow temperature | Reflow Profile per JEDEC J-STD-020D | - | 260 | ºC |
| $I_{LU}$ | Latch-up current | | - | 10 | mA |
| $V_{ESDHBM}$ | ESD Human Body Model (HBM) | | - | 2000 | V |
| $V_{ESDCDM}$ | ESD Charged Device Model (CDM) | | - | 250 | V |

[1] High side power switches are available on PAD3 and PAD4

[2] A low side power switch is available on PAD11

## 15.2 Recommended Operating Conditions

**Table 550: Recommended Operating Conditions[1]**

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| VDDP | Pad supply voltage | 1.8 | | 3.8 | V |
| VDDH | Digital supply voltage | 1.8 | | 3.8 | V |
| VDDA | Analog supply voltage | 1.8 | | 3.8 | V |
| $V_{INPUT}$ | Voltage on any input pin | 0 | | VDDH/VIO | V |
| $T_A$ | Ambient operating temperature | -40 | - | 85 | ºC |
| $F_{HFRC}$ | High Frequency RC oscillator frequency (HFRC) | | 24 | | MHz |
| $F_{LFRC}$ | Low Frequency RC oscillator frequency (LFRC) | | 1.024 | | kHz |
| $F_{XTAL}$ | Crystal frequency | | 32.768 | | kHz |

[1] $V_{DD}$ = VDDP = VDDA = VDDH

## 15.3 Current Consumption

**Table 551: Current Consumption**

| Symbol | Parameter | Test Conditions[1,2] | VDD | Typ | Unit |
|---|---|---|---|---|---|
| $I_{RUNF}$ | Flash program run current | Executing CoreMark from internal flash memory, HFRC = 24 MHz, all peripherals disabled, buck converters enabled | 3.3 V | 35 | μA/MHz |
| | | | 1.8 V | 51 | μA/MHz |
| $I_{RUNF}$ | Flash program run current | Executing CoreMark from internal flash memory, HFRC = 24 MHz, all peripherals disabled, buck converters disabled | 3.3 V | 79 | μA/MHz |
| | | | 1.8 V | 78 | μA/MHz |
| $I_{SLEEP}$ | Sleep mode current | WFI instruction with SLEEPDEEP = 0, buck converters enabled | 3.3 V | 57 | μA |
| | | | 1.8 V | 56 | μA |
| $I_{SLEEP}$ | Sleep mode current | WFI instruction with SLEEPDEEP = 0, buck converters disabled | 3.3 V | 63 | μA |
| | | | 1.8 V | 57 | μA |
| $I_{DEEPSLEEP}$ | Deep Sleep mode current | WFI instruction with SLEEPDEEP = 1 | 3.3 V | 143 | nA |
| | | | 1.8 V | 120 | nA |
| $I_{DEEPSLEEP}$ | Deep Sleep mode current with RTC active, clocked from 32.768 kHz crystal | WFI instruction with SLEEPDEEP = 1 | 3.3 V | 419 | nA |
| | | | 1.8 V | 390 | nA |
| $I_{DEEPSLEEP}$ with 8 KB of RAM | Deep Sleep mode current with 8 KB of RAM retained | WFI instruction with SLEEPDEEP = 1 | 3.3 V | 193 | nA |
| | | | 1.8 V | 170 | nA |
| $I_{DEEPSLEEP}$ with 16 KB of RAM | Deep Sleep mode current with 16 KB of RAM retained | WFI instruction with SLEEPDEEP = 1 | 3.3 V | 243 | nA |
| | | | 1.8 V | 220 | nA |
| $I_{DEEPSLEEP}$ with 32 KB of RAM | Deep Sleep mode current with 32 KB of RAM retained | WFI instruction with SLEEPDEEP = 1 | 3.3 V | 343 | nA |
| | | | 1.8 V | 320 | nA |
| $I_{DEEPSLEEP}$ with 64 KB of RAM | Deep Sleep mode current with 64 KB of RAM retained | WFI instruction with SLEEPDEEP = 1 | 3.3 V | 543 | nA |
| | | | 1.8 V | 520 | nA |

1. Core clock (HCLK) is 24 MHz for each parameter unless otherwise noted.
2. All values measured at 25°C

## 15.4 Power Mode Transitions

Table 552: Power Mode Transitions

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| **Buck mode** | | | | | | |
| $T_{RUN\_TO\_SLEEP}$ | Run to Sleep mode transition time | HCLK frequency = 24 MHz | - | 200 | 240 | ns |
| $T_{RUN\_TO\_DEEPSLEEP}$ | Run to Deep Sleep mode transition time | HCLK frequency = 24 MHz | - | 3.2 | 3.7 | μs |
| $T_{SLEEP\_TO\_RUN}$ | Sleep to Run mode transition time | | - | 450 | 575 | ns |
| $T_{DEEPSLEEP\_TO\_RUN}$ | Deep Sleep to Run mode transition time | | - | 37.5 | 43 | μs |
| **LDO mode** | | | | | | |
| $T_{RUN\_TO\_SLEEP}$ | Run to Sleep mode transition time | HCLK frequency = 24 MHz | - | 200 | 240 | ns |
| $T_{RUN\_TO\_DEEPSLEEP}$ | Run to Deep Sleep mode transition time | HCLK frequency = 24 MHz | - | 3.2 | 3.7 | μs |
| $T_{SLEEP\_TO\_RUN}$ | Sleep to Run mode transition time | | - | 450 | 575 | ns |
| $T_{DEEPSLEEP\_TO\_RUN}$ | Deep-Sleep to Run mode transition time | | - | 13.5 | 15.5 | μs |

## 15.5 Clocks/Oscillators

**Table 553: Clocks/Oscillators**

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $F_{HFRC}$ | HFRC frequency | | - | 24 | - | MHz |
| $F_{LFRC}$ | LFRC frequency | | - | 1024 | - | Hz |
| $F_{XT}$ | XT frequency | | - | 32.768 | - | kHz |
| $T_{J\_HFRC}$ | HFRC cycle-to-cycle jitter | 25ºC, STDEV of period | - | 100 | 230 | ps |
| $T_{J\_LFRC}$ | LFRC cycle-to-cycle jitter | 25ºC, STDEV of period | - | 6 | 8 | µs |
| $DC_{HFRC}$ | HFRC duty cycle | HFRC/8 CLKOUT | 40 | 50 | 60 | % |
| $DC_{LFRC}$ | LFRC duty cycle | 25ºC, LFRC/2 | - | 50 | - | % |
| $DC_{XT}$ | XT duty cycle | 25ºC | - | 45 | - | % |
| $C_{INX}$ | Internal XI/XO pin capacitance | | - | 2.3 | - | pF |
| $C_{EX}$ | External XI/XO pin capacitance | | - | - | 4.7 | pF |
| $F_{OF}$ | XT oscillator failure detection frequency | | - | 8 | - | kHz |
| $OA_{XT}$ | XT oscillation allowance | 32.768 kHz tuning fork crystal, no external XI/XO pin capacitance, TRIMCOREBXTAL = 0x2 | 400 | 700 | - | kΩ |

## 15.6 Analog-to-Digital Converter (ADC)

**Table 554: Analog to Digital Converter (ADC)**

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| **Analog Input** | | | | | | |
| $V_{ADCIN}$ | Input voltage range (ref = VDDP) | | 0 | - | VDDP | V |
| $I_{ADC0IN}$ | ADC channel 0 pin input leakage current (static) | | - | 5 | 10 | μA |
| $I_{ADC17IN}$ | ADC channels 1-7 pin input leakage current (static) | | - | 0.1 | 20 | nA |
| $C_{ADCIN}$ | Input source capacitance | | - | - | 10 | pF |
| $I_{ADC\_RUN}$ | Average run current | Single sample, single conversion, 1 slot enabled, 12 MHz ADC clock, total average SOC current and time duration (from ADC trigger event, conversion complete, then back into idle state), ADC power mode set to LPMODE 2 | - | 0.4 | - | μA |
| $I_{ADC\_RUN}$ | Average run current | Single sample, single conversion, 1 slot enabled, 12 MHz ADC clock, total average SOC current and time duration (from ADC trigger event, conversion complete, then back into idle state), ADC power mode set to LPMODE 1 | - | 9.3 | - | μA |
| **Sampling Dynamics** | | | | | | |
| $F_{ADCLK}$ | Conversion clock frequency | | 1.5 | - | 12 | MHz |
| $F_{ADCONV}$ | Conversion rate | | 115 | - | 800 | kSps |
| $T_{ADCSAMPLE}$ | Sample time[1] | | - | 15.5/ $F_{ADCLK}$ | 15/ $F_{ADCLK}$ | |
| $T_{TRIG\_TO\_START0}$ | Delay from trigger to start of conversion – LPM0, Slot 0 HFRC running | | - | 1/ $F_{ADCLK}$ | - | |
| $T_{TRIG\_TO\_START2}$ | Delay from trigger to start of conversion – LPM2, Slot 0 HFRC running | | - | 150 | - | μs |
| **Dynamic Characteristics** | | | | | | |
| $THD_{ADC}$ | Total harmonic distortion (THD), Single-ended input | LDO mode -<br>Single Ended Input | - | 0.05 | 0.12 | % |
| $SNR_{ADC}$ | Signal-to-noise ratio (SNR), Single-ended input | LDO mode -<br>Single Ended Input | 50 | 54 | - | dB |
| $SINAD_{ADC}$ | Signal-to-noise and distortion ration (SINAD), Single-ended input | LDO mode -<br>Single Ended Input | 50 | 54 | - | dB |
| $SFDR_{ADC}$ | Spurious-free dynamic range (SFDR), Single-ended input | LDO mode -<br>Single Ended Input | 62 | 67 | - | dB |

### Table 554: Analog to Digital Converter (ADC)

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $THD_{ADC}$ | Total harmonic distortion (THD), Single-ended input | Buck mode -<br>Single Ended Input | - | 0.06 | 0.12 | % |
| $SNR_{ADC}$ | Signal-to-noise ratio (SNR), Single-ended input | Buck mode -<br>Single Ended Input | 47 | 52 | - | dB |
| $SINAD_{ADC}$ | Signal-to-noise and distortion ration (SINAD), Single-ended input | Buck mode -<br>Single Ended Input | 47 | 52 | - | dB |
| $SFDR_{ADC}$ | Spurious-free dynamic range (SFDR), Single-ended input | Buck mode -<br>Single Ended Input | 62 | 67 | - | dB |
| **Performance** | | | | | | |
| $N_{ADC}$ | Resolution | | - | 10 | - | bits |
| $NMC_{ADC}$ | No missing codes | | 10 | - | - | bits |
| $INL_{ADC}$ | Integral nonlinearity Full input range | Full input range,<br>Channel 4 | - | 1.8 | - | LSB |
| $DNL_{ADC}$ | Differential nonlinearity Full input range | Full input range,<br>Channel 4 | - | 1 | - | LSB |
| $E_{ADC\_OFFEST}$ | Offset error | Full input range,<br>Channel 4 | - | 1 | - | LSB |
| $E_{ADC\_GAIN}$ | Gain error | Full input range,<br>Channel 4 | - | 0.3 | - | LSB |
| $E_{ADC\_TOTAL}$ | Total unadjusted error Full input range | Full input range,<br>Channel 4 | - | 1.3 | - | LSB |
| **Internal Temperature Sensor** | | | | | | |
| $E_{TEMP}$ | Temperature sensor accuracy | After calibration | -4 | - | 4 | °C |
| $S_{TEMP}$ | Temperature sensor slope | | - | 3.57 | - | mV/°C |
| $V_{TEMP25C}$ | Temperature sensor voltage reading, when junction temperature is at 25° C | | - | 1.065 | - | V |
| **Internal Reference** | | | | | | |
| $V_{ADCREFIN}$ | Voltage range | | 1.495 | 1.5 | 1.505 | V |
| $N_{REF}$ | Reference resolution | | - | 11 | - | bits |
| $T_{REFSETTLE}$ | Settling time | | - | - | 80 | ns |
| $I_{ADCREF\_IN}$ | Internal reference current | When reference is enabled for sampling | - | 25 | - | µA |
| **External Reference** | | | | | | |
| $V_{ADCREFEXT}$ | Voltage range | | 1.495 | 1.5 | VDDA | V |
| $I_{ADCREF\_IN}$ | ADC external reference pin input leakage current | | - | - | - | nA |
| **Battery Resistance** | | | | | | |
| $BAT_{RES}$ | Internal resistance for Batter Measurement | VDD = 3.8 V | 484 | 510 | 540 | Ω |

1. Max sample time is for less than or equal to eight samples. Typical sample rate is the average over greater than eight samples.

## 15.7 Buck Converter

Table 555: Buck Converter

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $L_{BUCK}$ | Buck converter inductance (VOUT1, VOUT2) | | - | 2.2 | - | µH |
| $C_{BUCK\_VOUT1}$ | Buck converter output capacitance (VOUT1) | | - | 2 | - | µF |
| $C_{BUCK\_VOUT2}$ | Buck converter output capacitance (VOUT2) | | - | 1 | - | µF |
| $V_{OUT1}$ | VOUT1 output voltage | | - | 0.77 | - | V |
| $V_{OUT2}$ | VOUT2 output voltage | | - | 1.15 | - | V |

## 15.8 Power-On RESET (POR) and Brown-Out Detector (BOD)

**Table 556: Power-On Reset (POR) and Brown-Out Detector (BOD)**

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $T_{VDDFS}$ | Falling slew rate on VDDP/ VDDA / VDDH | - | - | 1.43 | V/ms |
| $V_{POR\_RISING}$ | POR rising threshold voltage | 1.72 | - | 2.05 | V |
| $V_{POR\_FALLING}$ | POR falling threshold voltage | 1.59 | 1.72 | 1.88 | V |
| $V_{POR\_HYS}$ | POR voltage hysteresis | - | 135 | - | mV |
| $V_{BOD1\_RISING}$ | BOD1 rising threshold voltage | - | 1.9 | - | V |
| $V_{BOD1\_FALLING}$ | BOD1 falling threshold voltage | 1.67 | 1.74 | 1.81 | V |
| $V_{BOD1\_HYS}$ | BOD1 voltage hysteresis | 100 | 150 | - | mV |
| $V_{BOD2\_RISING}$ | BOD2 rising threshold voltage | - | 2.5 | - | V |
| $V_{BOD2\_FALLING}$ | BOD2 falling threshold voltage | 2.1 | 2.31 | 2.5 | V |
| $V_{BOD2\_HYS}$ | BOD2 voltage hysteresis | - | 80 | - | mV |

## 15.9 Resets

Table 557: Resets

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $T_{RST}$ | nRST pulse width to guarantee reset assertion | - | 1.3 | - | µs |
| $T_{POR}$ | POR detect to nRST deassertion delay | 170 | 180 | 230 | ms |
| $T_{POR2HRST}$ | Delay from nRST deassertion to HRESET deassertion | 2.3 | - | 2.5 | ms |
| $T_{HWRST}$ | Delay from nRST assertion to HRESET assertion | 420 | - | 480 | µs |
| $T_{HWRST}$ | Delay from nRST deassertion to HRESET deassertion | 410 | - | 520 | µs |
| $T_{RSTDLY}$ | nRST reset delay from internal BODL | - | 30 | - | µs |
| $T_{SOFT}$ | Software initiated reset delay | 320 | - | 660 | µs |

## 15.10 Voltage Comparator (VCOMP)

Table 558: Voltage Comparator (VCOMP)

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{COMPIN}$ | Input voltage range 0 | | - | VDDP | - | V |
| $V_{COMPIN\_OV}$ | Input offset voltage | | - | 34 | - | mV |
| $I_{COMPIN\_LEAK}$ | Input leakage current | COMPIN1 | - | 1 | 20 | nA |
| $V_{COMP\_FTHR}$ | Input rising threshold voltage | | - | 1.57 | - | V |
| $V_{COMP\_RTHR}$ | Input falling threshold voltage | | - | 1.5 | - | V |
| $T_{COMP\_RTRIG}$ | Rising voltage trigger response time | | - | 25 | - | μs |
| $T_{COMP\_FTRIG}$ | Falling voltage trigger response time | | - | 20 | - | μs |
| $V_{HYST}$ | Hysteresis | | - | 75 | - | mV |

## 15.11 Internal DAC Reference for VCOMP

Table 559: Internal DAC Reference for VCOMP

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $N_{DACRES}$ | Resolution |  | - | 4 | - | bits |

## 15.12 Inter-Integrated Circuit ($I^2C$) Interface

Table 560: Inter-Integrated Circuit ($I^2C$) Interface

| Symbo l | Parameter | Test Conditions | VDD | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| $f_{SCL}$ | SCL clock frequency | | 1.8 V - 3.8 V | 10 | - | 1000 | kHz |
| $t_{LOW}$ | Low period of SCL clock | | 1.8 V - 3.8 V | 1.3 | - | - | μs |
| $t_{HIGH}$ | High period of SCL clock | | 1.8 V - 3.8 V | 600 | - | - | ns |
| $t_{RISE}$ | Rise time of SDA and SCL | | 1.8 V - 3.8 V | - | - | 300 | ns |
| $t_{FALL}$ | Fall time of SDA and SCL | | 1.8 V - 3.8 V | - | - | 300 | ns |
| $t_{HD:STA}$ | START condition hold time | | 1.8 V - 3.8 V | 600 | - | - | ns |
| $t_{SU:STA}$ | START condition setup time | | 1.8 V - 3.8 V | 600 | - | - | ns |
| $t_{SU:DAT}$ | SDA setup time | | 1.8 V - 3.8 V | 100 | - | - | ns |
| $t_{HD:DAT}$ | SDA hold time | | 1.8 V - 3.8 V | 0 | - | - | ns |
| $t_{VD:DAT}$ | SDA data valid time | 1.5 kΩ internal pull-up resistor and 100 pF total SDA capacitive loading. | 1.8 V - 3.8 V | - | - | 450 | ns |
| $t_{SU:STO}$ | STOP condition setup time | | 1.8 V - 3.8 V | 600 | - | - | ns |
| $t_{BUF}$ | Bus free time before a new transmission | | 1.8 V - 3.8 V | 1.3 | - | - | μs |

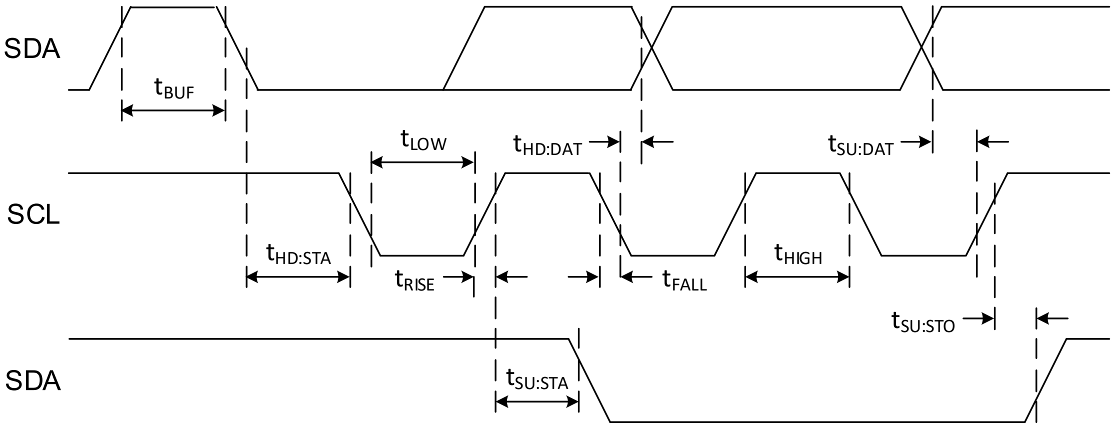


Figure 54. $I^2C$ Timing

## 15.13 Serial Peripheral Interface (SPI) Master Interface

**Table 561: Serial Peripheral Interface (SPI) Master Interface**

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $F_{SCLK}$ | SCLK frequency range | .002 | 8 | 8 | MHz |
| $B_{FIFO}$ | FIFO size | 64 | 64 | 64 | Bytes |
| $T_{SCLK\_LO}$ | Clock low time | $1/2F_{S-CLK(max)}$ | 1/2 FSCLK | - | s |
| $T_{SCLK\_HI}$ | Clock high time | $1/2F_{S-CLK(max)}$ | 1/2 FSCLK | - | s |
| $T_{SU\_MI}$ | MISO input data setup time | - | - | 0.8 | ns |
| $T_{HD\_MI}$ | MISO input data hold time | 4.2 | - | - | ns |
| $T_{VALID,MO}$ | MOSI output data valid time | 15.6 | - | - | ns |

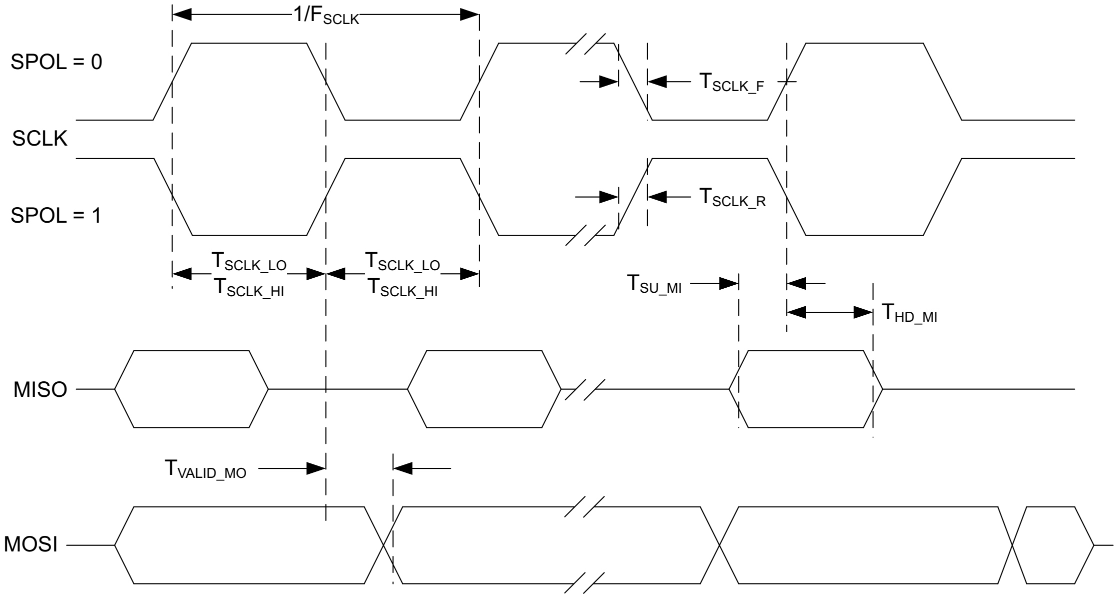


**Figure 55. SPI Master Mode, Phase = 0**

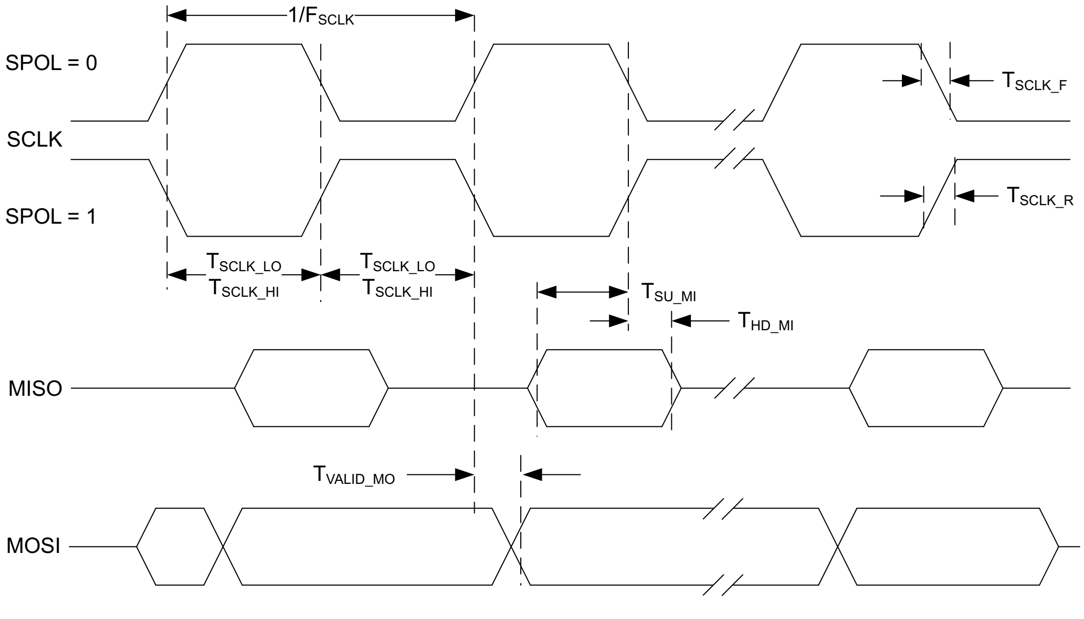


**Figure 56. SPI Master Mode, Phase = 1**

## 15.14 Serial Peripheral Interface (SPI) Slave Interface

Table 562: Serial Peripheral Interface (SPI) Slave Interface

| Symbol | Parameter | Test Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $F_{SCLK}$ | SCLK frequency range | | - | - | 24 | MHz |
| $B_{FIFO}$ | FIFO size | | 64 | 64 | 64 | Bytes |
| $T_{SCLK\_LO}$ | Clock low time | | $1/2F_{S\text{-}CLK(max)}$ | 1/2 FSCLK | - | s |
| $T_{SCLK\_HI}$ | Clock high time | | $1/2F_{S\text{-}CLK(max)}$ | 1/2 FSCLK | - | s |
| $T_{CE\_LEAD}$ | Chip enable low to first SCLK edge | | - | 10 | - | ns |
| $T_{CE\_LAG}$ | Chip enable high to last SCLK edge | | - | 10 | - | ns |
| $T_{CE\_SDO}$ | Chip enable low to MISO data output | | - | 5 | - | ns |
| $T_{CE\_SDZ}$ | Chip enable high to MISO data tri-state | | - | 5 | - | ns |
| $T_{SU\_SI}$ | MOSI input data setup time | SPOL = 0 | - | - | 0.8 | ns |
| $T_{HD\_SI}$ | MOSI input data hold time | SPOL = 0 | 4.2 | - | - | ns |
| $T_{VALID,SO}$ | MISO output data valid time | SPOL = 0 | 15.6 | - | - | ns |
| $T_{SU\_SI}$ | MOSI input data setup time | SPOL = 1 | - | - | 0.4 | ns |
| $T_{HD\_SI}$ | MOSI input data hold time | SPOL = 1 | 4.2 | - | - | ns |
| $T_{VALID,SO}$ | MISO output data valid time | SPOL = 1 | 14.8 | - | - | ns |

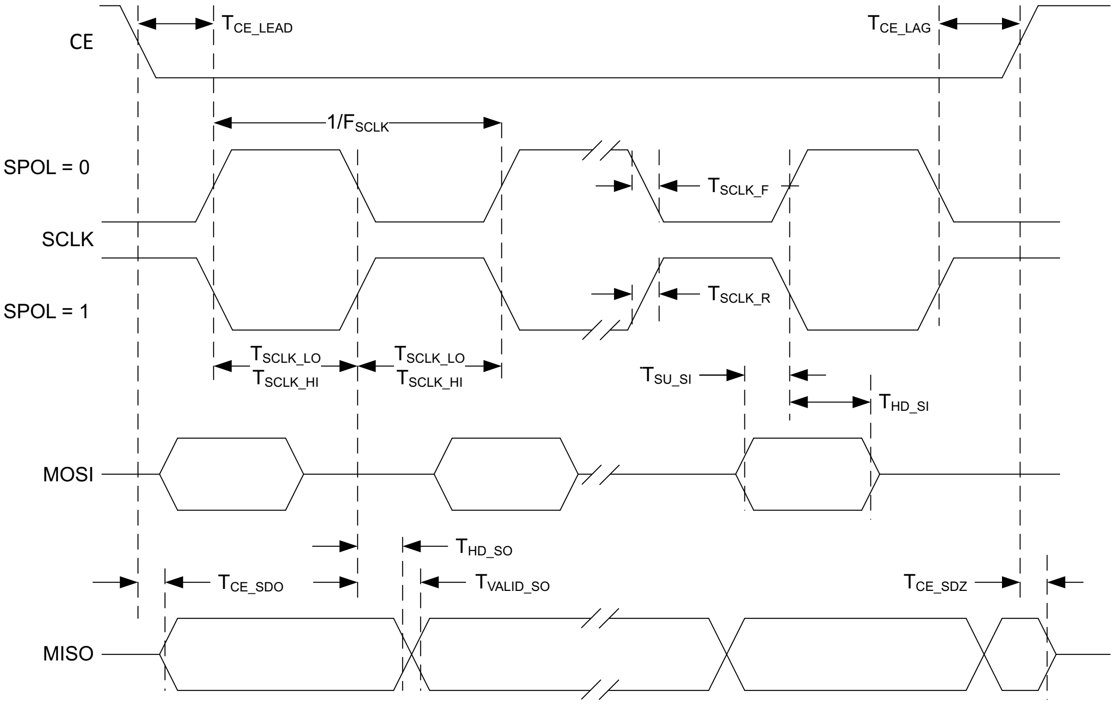


**Figure 57. SPI Slave Mode, Phase = 0**

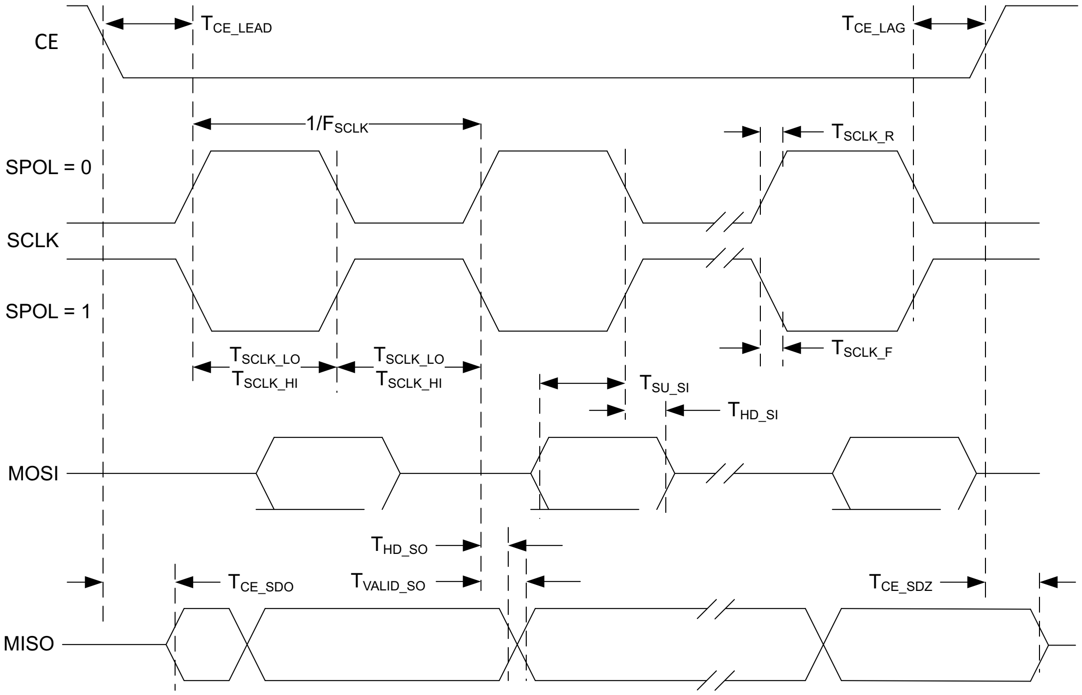


**Figure 58. SPI Slave Mode, Phase = 1**

## 15.15 Universal Asynchronous Receiver/Transmitter (UART)

**Table 563: Universal Asynchronous Receiver/Transmitter (UART)**

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $F_{BAUD}$ | UART baud rate | - | - | 921600 | bps |

## 15.16 Counter/Timer (CTIMER)

Table 564: Counter/Timer (CTIMER)

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $F_{CTIMER}$ | External clock frequency | - | - | 24 | MHz |
| $T_{CTIMER}$ | External clock pulse width | 100 | - | - | ns |

## 15.17 Flash Memory

**Table 565: Flash Memory**

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $PE_{CYC}$ | Program/erase cycles before failure | 10K | - | - | cycles |
| $T_{FDR}$ | Data retention | 10 | - | - | years |
| $T_{PAGE\_ERASE}$ | Single page erase time (2048 bytes) | - | 40 | - | ms |
| $T_{MASS\_ERASE}$ | Mass erase time | - | 40 | - | ms |
| $I_{PROGRAM}$ | Supply current during a page program | - |  | 4.3 | mA |
| $I_{ERASE}$ | Supply current during a page erase | - | - | 3.44 | mA |

## 15.18 General Purpose Input/Output (GPIO)

All GPIOs have Schmitt trigger inputs.

**Table 566: General Purpose Input/Output (GPIO)**

| Symbol | Parameter | Condition | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| **All GPIOs** | | | | | | |
| $V_{OH}$ | High-level output voltage | 3.0 mA load, low drive strength | 0.8 * VDDP | -- | -- | V |
| | | 5.7 mA load, high drive strength | | | | |
| $V_{OL}$ | Low-level output voltage | 2.6 mA load, low drive strength | - | - | 0.2 * VDDP | V |
| | | 5.1mA load, high drive strength | | | | |
| $V_{IH}$ | Positive going input threshold voltage | | 0.61*VDDP | - | - | V |
| $V_{IL}$ | Negative going input threshold voltage | | - | - | 0.33*VDDP | V |
| $V_{HYS}$ | Input hysteresis | | 0.225 | 0.325 | 0.425 | V |
| $C_{GPI}$ | Input capacitance | | - | 3.6 | - | pF |
| $R_{PU}$ | Pull-up resistance | | 25 | 40 | 180 | kΩ |
| $R_{PD}$ | Pull-down resistance | | 25 | 65 | 800 | kΩ |
| $R_{PUI2C00}$ | $I^2C$ pad pull-up resistance, RSEL = 0x00 | | - | 1.5 | - | kΩ |
| $R_{PUI2C01}$ | $I^2C$ pad pull-up resistance, RSEL = 0x01 | | - | 6 | - | kΩ |
| $R_{PUI2C10}$ | $I^2C$ pad pull-up resistance, RSEL = 0x10 | | - | 12 | - | kΩ |
| $R_{PUI2C11}$ | $I^2C$ pad pull-up resistance, RSEL = 0x11 | | - | 24 | - | kΩ |
| $I_{IN}$ | Input pin leakage current | | - | 1 | 20 | nA |
| $I_{INOD}$ | Open drain output leakage current | | - | 1 | 20 | nA |
| **Standard GPIOs** | | | | | | |
| $I_{SRC\_STD}$ | Output source current | VDDP = 1.8V, low drive strength | 3 | 4.4 | - | mA |
| $I_{SRC\_STD}$ | Output source current | VDDP = 1.8V, high drive strength | 5.7 | 8.4 | - | mA |
| $I_{SNK\_STD}$ | Output sink current | VDDP = 1.8V, low drive strength | 2.6 | 4 | - | mA |
| $I_{SNK\_STD}$ | Output sink current | VDDP = 1.8V, high drive strength | 5.1 | 7.7 | - | mA |
| **Power Switch GPIOs** | | | | | | |
| $PWR_{SOURCE\_RES}$ | High side power switch resistance | | - | 1.3 | 2 | Ω |

**Table 566: General Purpose Input/Output (GPIO)**

| | | | | | | |
|---|---|---|---|---|---|---|
| $I_{SRC\_PWR}$ | High side power switch source current | | 50 | - | - | mA |
| $PWR_{SINK\_RES}$ | Low side power switch resistance | | - | 1.3 | 3 | Ω |
| $I_{SNK\_PWR}$ | Low side power switch sink current | | 50 | - | - | mA |

## 15.19 Serial Wire Debug (SWD)

**Table 567: Serial Wire Debug (SWD)**

| Symbol | Parameter | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| $T_{SWDCK\_HI}$ | SWDCK clock high period | 0.085 | - | 500 | μs |
| $T_{SWDCK\_LO}$ | SWDCK clock low period | 0.085 | - | 500 | μs |
| $T_{OS\_SWD}$ | SWDIO output skew to falling edge of SWDCLK | -5 | - | 45 | ns |
| $T_{SU\_SWD}$ | Input setup time between SWDIO and rising edge SWDCK | 65 | - | - | ns |
| $T_{HD\_SWD}$ | Input hold time between SWDIO and rising edge SWDCK | 125 | - | - | ns |

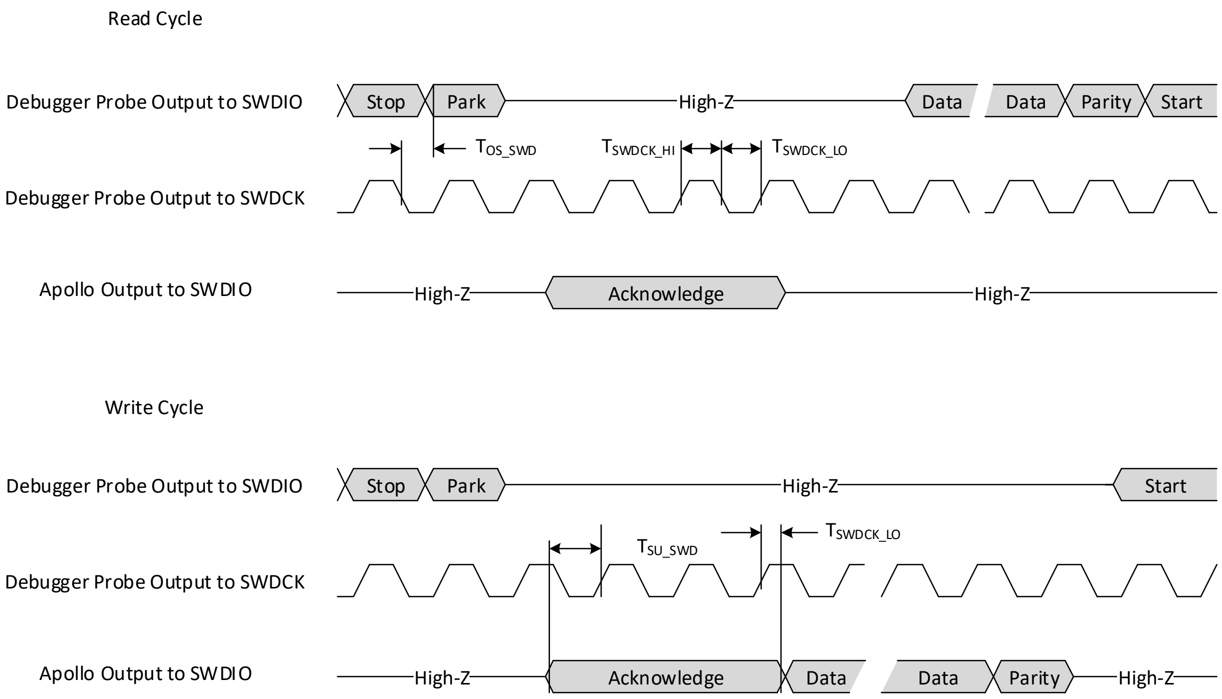
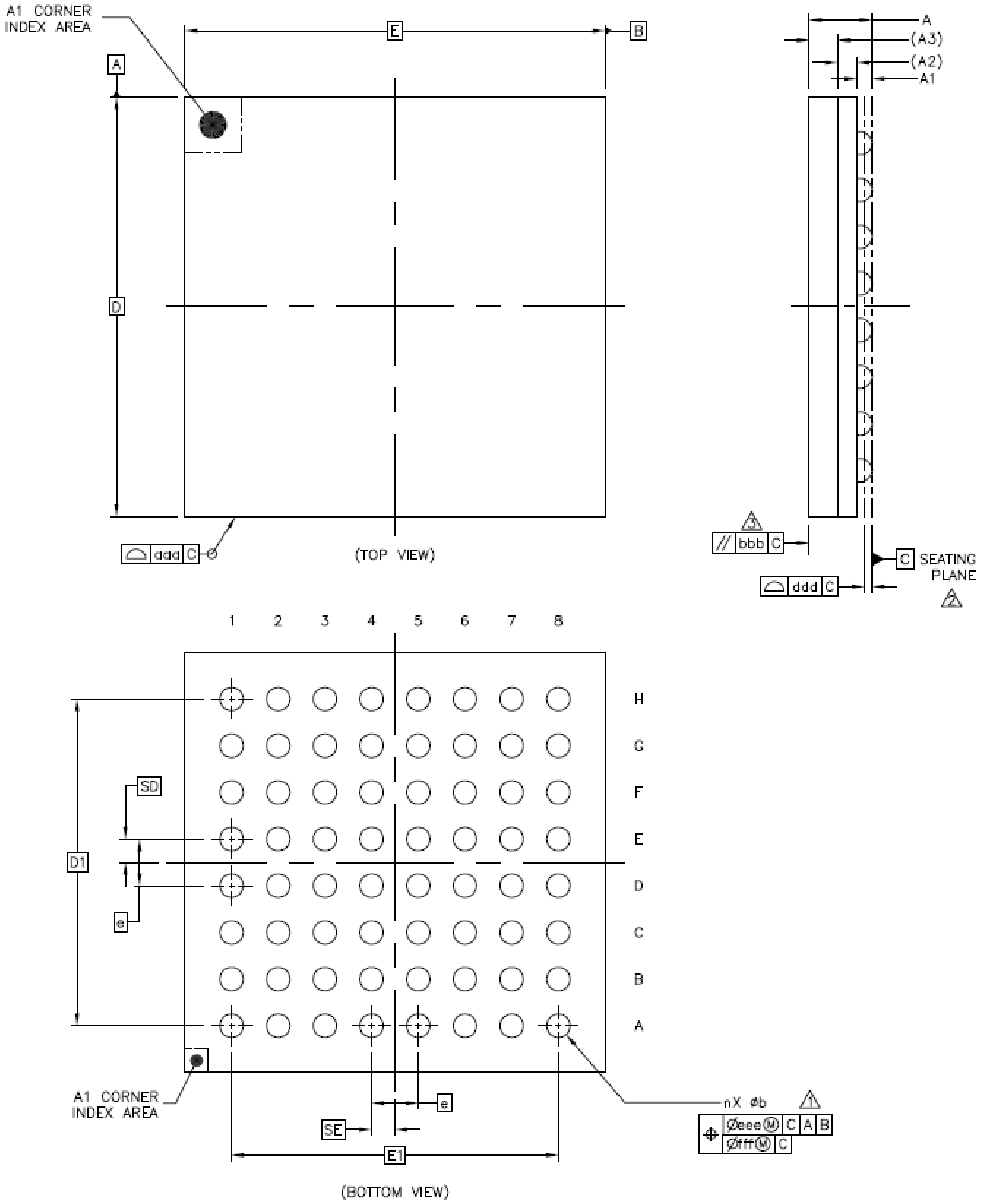


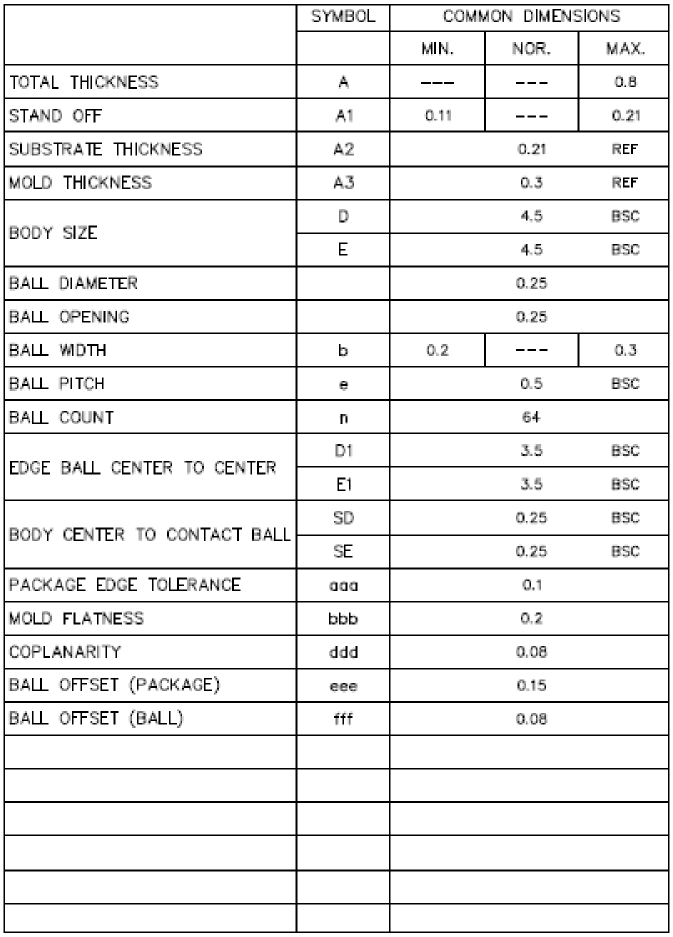
**Figure 59. Serial Wire Debug Timing**

# 16. Package Mechanical Information

## 16.1 BGA Package[1]



[1] All dimensions in mm unless otherwise noted.



| | SYMBOL | COMMON DIMENSIONS | | |
|---|---|---|---|---|
| | | MIN. | NOR. | MAX. |
| TOTAL THICKNESS | A | --- | --- | 0.8 |
| STAND OFF | A1 | 0.11 | --- | 0.21 |
| SUBSTRATE THICKNESS | A2 | | 0.21 | REF |
| MOLD THICKNESS | A3 | | 0.3 | REF |
| BODY SIZE | D | | 4.5 | BSC |
| | E | | 4.5 | BSC |
| BALL DIAMETER | | | 0.25 | |
| BALL OPENING | | | 0.25 | |
| BALL WIDTH | b | 0.2 | --- | 0.3 |
| BALL PITCH | e | | 0.5 | BSC |
| BALL COUNT | n | | 64 | |
| EDGE BALL CENTER TO CENTER | D1 | | 3.5 | BSC |
| | E1 | | 3.5 | BSC |
| BODY CENTER TO CONTACT BALL | SD | | 0.25 | BSC |
| | SE | | 0.25 | BSC |
| PACKAGE EDGE TOLERANCE | aaa | | 0.1 | |
| MOLD FLATNESS | bbb | | 0.2 | |
| COPLANARITY | ddd | | 0.08 | |
| BALL OFFSET (PACKAGE) | eee | | 0.15 | |
| BALL OFFSET (BALL) | fff | | 0.08 | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Drawing Notes:
Ball width is measured at maximum solder ball diameter, parallel to seating plane.
Seating plane is defined by spherical crowns of solder balls.

Parallelism measurement shall exclude effect of mark on top surface of package.

**Figure 60. BGA Package Drawing**

### 16.1.1 PCB land pattern and solder stencil

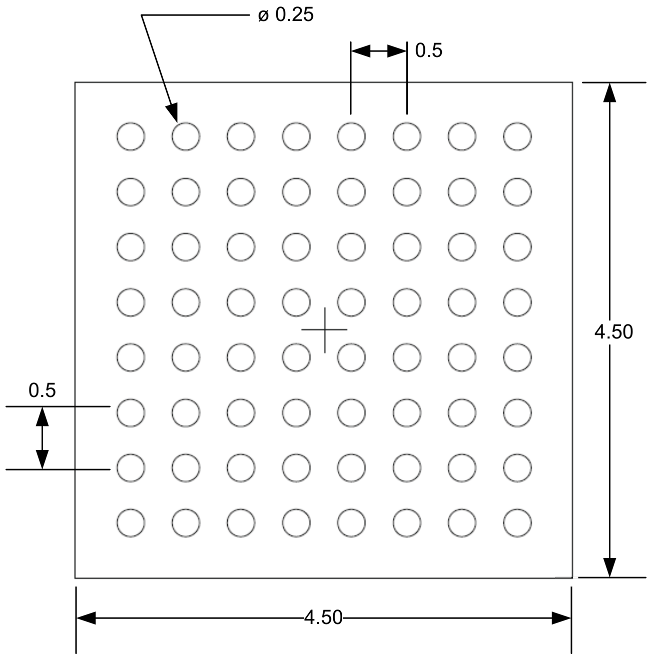


**Figure 61. Example PCB Land Pattern for BGA Package**

A 4 mil stencil is recommended.

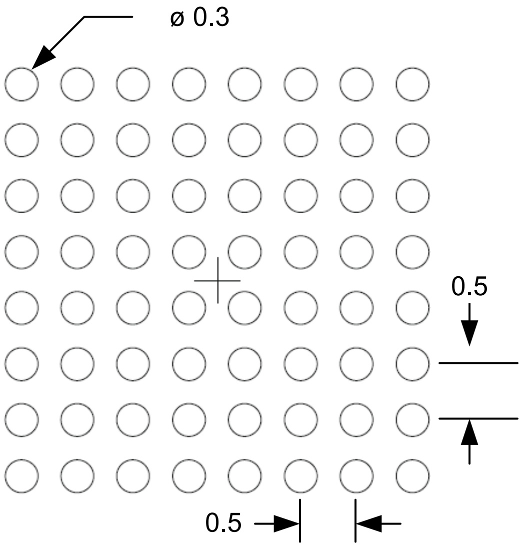
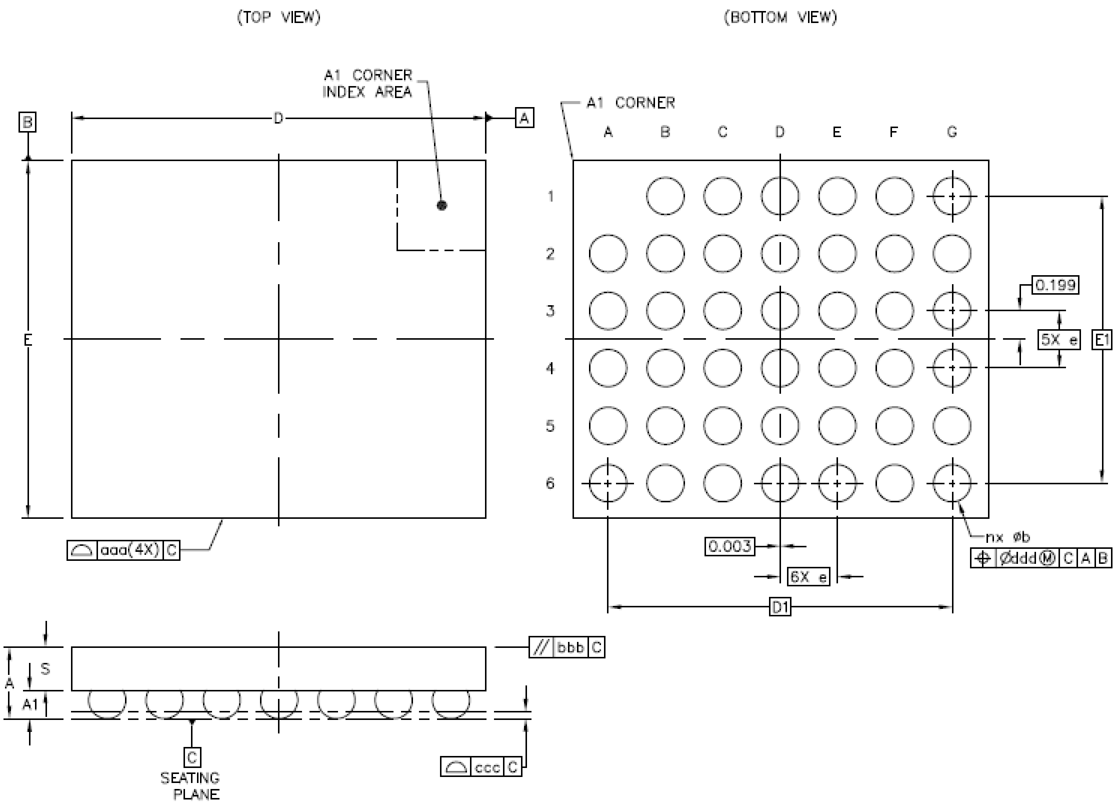


**Figure 62. Example Solder Stencil Pattern for BGA Package**

## 16.2 CSP Package



| | | SYMBOL | COMMON DIMENSIONS |
|---|---|---|---|
| TOTAL THICKNESS | | A | 0.5 ± 0.055 |
| WAFER THICKNESS | | S | 0.3 ± 0.025 |
| STAND OFF | | A1 | 0.17 ~ 0.23 |
| FILM THICKNESS | | A2 | --- ± --- |
| BODY SIZE | X | D | 2.9 |
| | Y | E | 2.49 |
| BALL/BUMP PITCH | X | SD | --- BSC |
| | Y | SE | --- BSC |
| EDGE BALL CENTER TO CENTER | X | D1 | 2.4 BSC |
| | Y | E1 | 2 BSC |
| PITCH | | e | 0.4 BSC |
| BALL DIAMETER (SIZE) | | | 0.25 |
| BALL/BUMP WIDTH | | b | 0.23 ~ 0.29 |
| BALL/BUMP COUNT | | n | 41 |
| PACKAGE EDGE TOLERANCE | | aaa | 0.03 |
| WAFER FLATNESS | | bbb | 0.06 |
| COPLANARITY | | ccc | 0.05 |
| BALL/BUMP OFFSET (PACKAGE) | | ddd | 0.015 |

**Figure 63. CSP Package Drawing**

## 16.3 Reflow Profile

Figure 64 illustrates the reflow soldering requirements.

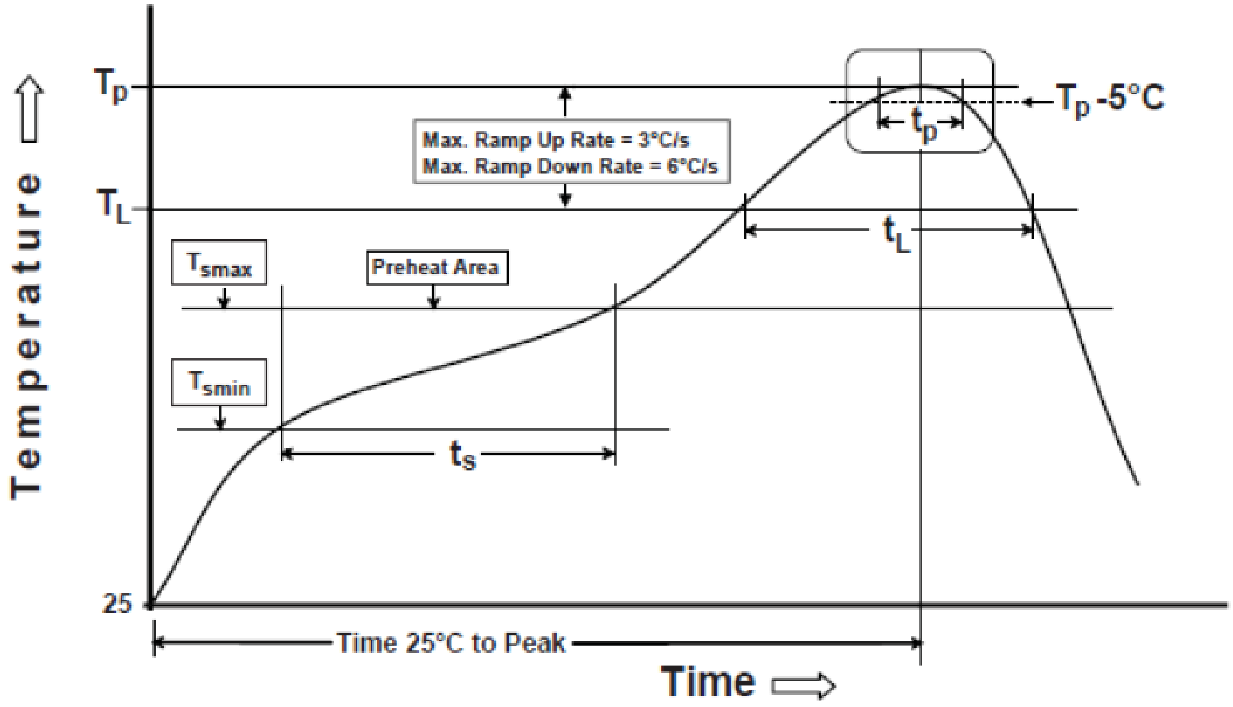


**Figure 64. Reflow Soldering Diagram**

**Table 568: Reflow Soldering Requirements (Pb-free assembly)**

| Profile Feature | Requirement |
|---|---|
| Preheat/Soak<br>Temperature Min (Tsmin)<br>Temperature Max (Tsmax)<br>Time (ts) from (Tsmin to Tsmax) | <br>150°C<br>200°C<br>60-120 seconds |
| Ramp-up rate (TL to Tp) | 3°C/second max. |
| Liquidous temperature (TL)<br>Time (tL) maintained above TL | 217°C<br>60-150 seconds |
| Peak package body temperature (Tp) | 260°C max. |
| Time (tp) within 5°C of Tp | 30 seconds max. |
| Ramp-down rate (Tp to TL) | 6°C/second max. |
| Time 25°C to peak temperature | 8 minutes max. |

## Index

Numerics
10BIT 89
12/24 Hour Mode 247
3-wire mode 92
A
Access permissions 41
accumulation control 316
accumulation, automatic 316
ACK 89, 116
Acknowledge 89
Active Mode 41
ADC 35, 157, 313
ADC Configuration Register 319
ADC_DIV3 314
ADC_EXT0 314
ADC_EXT1 314
ADC_EXT2 314
ADC_EXT3 314
ADC_EXT4 314
ADC_EXT5 314
ADC_EXT6 314
ADC_EXT7 314
ADCREF 157
ADC_SWT 315
ADC_TEMP 314
ADC_TRIG0 315
ADC_TRIG1 315
ADC_TRIG2 315
ADC_TRIG5 315
ADC_TRIG6 315
ADC_TRIG7 315
ADC_VDD 314
ADC_VSS 314
AHB 41, 110
Alarm Registers 247
ALM interrupt 247
ALM100 247
AM08XX, *see* real-time clock
AM18XX, *see* real-time clock
AMBA 41
Analog Multiplexer Channel Selection 316
analog mux 314
analog-to-digital converter 35
APB 41

Architecture Reference Manual 42
Area, Direct 111
ARM, *see* processor
ARMv7 41
Autoadjustment, HFRC 230
Automatic Sample Accumulation 316
B
Bandgap 315
battery life 34
BCD format 246
Buck Converter Charge Insertion 260
Buck Converters 354
Bus Not Busy 88, 116
bus, AMBA AHB 41
bus, AMBA APB 41
bus, DCode 41
bus, ICode 41
bus, System 41
C
Calibration, Distributed Digital 229, 230
CALRC 229
CALXT 230
CLKOUT 228, 229, 231
Clock Generator Module 228
clock sources 35
clock, interface 87
CMPOUT 346
CMPR0 258, 259
CMPR1 259
comparator, window 313
CONT 92
Continuous 259
control, accumulation 316
converters, buck 354
core, *see* processor
Cortex, *see* processor
counter, 32-bit 260
CPHA 95, 121
CPOL 95, 121
CTRERR 246
CTS 157
D
Data Valid 89, 116
DCode 41
debug 43
Debug Interfaces 43

Deep Sleep 229
Deep Sleep Mode 41, 42
Deep Slope Mode 121
Direct Area 111
Distributed Digital Calibration 229, 230
E
EG_IOSLAVE_FIFOPTR_FIFOSIZ 112
event 36
event, wakeup 41
F
Fast Mode Plus 115
fault handler, Memanage 41
Faulting Address Trapping Hardware 43
FIFO 110, 313
    Area Functions 112
FIFOCTR 113
FIFOPTR 112
FIFOREM 88
FIFORTHR 88
FIFOSIZ 88, 112
FIFOUPD 114
FIFOWTHR 88
flash 35
G
GPIO 144
GPIO and Pad Configuration Module 140
GPIOA_IER 145
GPIOA_ISR 145
GPIOA_WCR 145
GPIOB_WSR 145
GPIOEN 144
GPIOENA 144
GPIOENB 144
GPIOnINCFG 144, 147
GPIOn_INT 147
GPIOnINTD 147
GPIOnINTP 144
GPIOnOUTCFG 146
GPIORD 147
GPIORDA 144
GPIORDB 144
GPIOWT 144, 145
GPIOWTA 144
GPIOWTB 144
GPIOWTCA 144
GPIOWTSA 144

H
Hardware, Fault Address Trapping 43
HFADJ Register 239
HFADJCK 231
HFRC 230, 314
HFRC Autoadjustment 230
HFRCDEL 232
HFXTADJ 231
High Frequency RC Oscillator 230
HR1224 247
I
I2C
- 10-bit addressing 117
- 7-bit addressing 117
- ADDRESS 89
- Address 117
- Command 89
- FIFO 88
- I2CADDR 115, 117
- IO Master 1 147, 148
- IO Master 1 Loopback 152
- IO Slave 151
- master 86
- Multi-master Arbitration 92
- Normal Read 91
- Normal Write 90
- Offset Address 90, 117
- Raw Read 91
- Raw Write 91
- Read 118
- receiver 88
- SCL 88
- SDA 88
- Slave 109, 115
- transmitter 88
- Write 118

I2C/SPI Master 228
I2C/SPI Master Module 86
ICode 41
IE bit 257
Instrumentation Trace Macrocell 43
Instrumentation Trace Module 35
interfaces, debug 43
interrupt 36
- ALM 247
- IOINT 114

RDERR 114
Interrupts
Vector Table 36
IO Slave Interrupt 152
IOINT 114
IOREAD 114
ITM, *see* Instrumentation Trace Macrocell
L
LDO 354
LFRC 229, 232
life, battery 34
Loopback 152
Low-Power Consumption Modes 36
M
Managed Conversion Slots 315
map, memory 38
Master Module , I2C/SPI 86
MCU_CTRL 232
MemManage 41
memory 35
LRAM 88, 110
RAM 35
SRAM 41
memory map 38
peripheral device 40
MISO 92
mode
Active 41
Deep Sleep 42, 229
Sleep 42
mode, Deep Sleep 41
Module, ADC and Temperature Sensor 313
module, PINCFG 92
MOSI 92
MPU, *see* Memory Protection Unit
mux, analog 314
N
NAK 116
Nested Vectored Interrupt Controller 36
NSEL 346
NVIC, *see* Nested Vectored Interrupt Controller
O
OPER 90
oscillator
High Frequency RC 230
high frequency RC 35, 228, 314

low frequency RC 35, 228, 229
RC 35
XTAL 35, 228, 229
XTAL Failure Detection 232
OUTDATSEL 145
OUTENSEL 145, 146, 147
P
PAD10PULL 149
PAD16FNCSEL 157
PAD20FNCSEL 159
PAD20INPEN 159, 160
PAD21INPEN 159
PAD5INPEN 147, 148
PAD5PULL 147
PAD5RSEL 147
PAD6INPEN 147, 148
PAD6PULL 147
PAD6RSEL 147
PAD8INPEN 147, 148, 149
PAD8PULL 147, 148, 149
PAD9INPEN 147, 148, 149
PAD9PULL 147, 148, 149
PADKEY 144
PADnFNCSEL 140–??, 142, 144
PADnINPEN 147, 150, 157
PADnPULL 141, 147, 157
PADnRSEL 147
Peripheral Device Memory Map 40
PINCFG 92
power 34
power modes 41
processor 35
Protected Memory System Architecture 40
Protection regions 40
PSEL 346
PTAT 346
PWDKEY 346
R
rate, sampling 313
RDERR 114
Real Time Clock Module 231
real-time clock 35
REG_CLK_GEN_ALMLOW_ALM100 247
REG_CLK_GEN_ALMLOW_ALMHR 247
REG_CLK_GEN_ALMLOW_ALMMIN 247
REG_CLK_GEN_ALMLOW_ALMSEC 247

REG_CLK_GEN_ALMUP_ALMDATE 247
REG_CLK_GEN_ALMUP_ALMMO 247
REG_CLK_GEN_ALMUP_ALMWKDY 247
REG_CLKGEN_CALXT 230
REG_CLK_GEN_CTRLOW_CTR100 246
REG_CLK_GEN_CTRLOW_CTRHR 246
REG_CLK_GEN_CTRLOW_CTRMIN 246
REG_CLK_GEN_CTRLOW_CTRSEC 246
REG_CLK_GEN_CTRUP_CB 246
REG_CLK_GEN_CTRUP_CEB 247
REG_CLK_GEN_CTRUP_CTRDATE 246
REG_CLK_GEN_CTRUP_CTRERR 246
REG_CLK_GEN_CTRUP_CTRMO 246
REG_CLK_GEN_CTRUP_CTRWKDY 246
REG_CLK_GEN_CTRUP_CTRYR 246
REG_CLKGEN_HFTUNERB 231
REG_CLKGEN_OCTRL_OSEL 229, 232
REG_CLK_GEN_RTCCTL_HR1224 247
REG_CLK_GEN_RTCCTL_RPT 247
REG_CLK_GEN_RTCCTL_RSTOP 246
REG_CLK_GEN_RTCCTL_WRTC 246
REG_CLKGEN_STATUS_OMODE 232
REG_CLKGEN_STATUS_OSCF 232
REG_CTIMER_CMPR0 257
REG_CTIMER_CMPR0/1 256
REG_CTIMER_CTCTRLx_CTRLINKx 260
REG_CTIMER_CTCTRLx_TMRxyFN 256
REG_CTIMER_TMRxyCLR 257
REG_CTIMER_TMRxyEN 258
REG_CTIMER_TMRxyIE 258
REG_CTIMER_TMRxyPE 257, 258
REG_CTIMER_TMRxyPOL 258
REG_GPIO_GPIOCFGy 144
REG_GPIO_GPIOyCFG_GPIOnOUTCFG 144
REG_GPIO_PADKEY 140
REG_GPIO_PADREG 140
REG_GPIO_PADREGy_PAD11PWRDN 141
REG_GPIO_PADREGy_PAD20PULL 140
REG_GPIO_PADREGy_PAD3PWRUP 141
REG_GPIO_PADREGy_PAD4PWRUP 141
REG_GPIO_PADREGy_PADnFNCSEL 140
REG_GPIO_PADREGy_PADnINPEN 141
REG_GPIO_PADREGy_PADnPULL 140
REG_GPIO_PADREGy_PADnSTRNG 140
REG_IOMSTRn_CLKCFG_DIV3 87
REG_IOMSTRn_CLKCFG_DIVEN 87

REG_IOMSTRn_CLKCFG_FSEL 87
REG_IOMSTRn_CLKCFG_LOWPER 87
REG_IOMSTRn_CLKCFG_TOTPER 87
REG_IOMSTRn_CMD 87
REG_IOMSTRn_FIFOPTR_FIFOREM 88
REG_IOMSTRn_FIFOPTR_FIFOSIZ 88
REG_IOMSTRn_FIFOTHR_FIFOWTHR 87
REG_IOMSTRn_IOMCFG_SPHA 92
REG_IOMSTRn_IOMCFG_SPOL 92
REG_IOSLAVE_FIFOCFG_FIFOBASE 110
REG_IOSLAVE_FIFOCFG_FIFOMAX 110
REG_IOSLAVE_FIFOCFG_ROBASE 112
REG_IOSLAVE_FIFOCTR 113
REG_IOSLAVE_FIFOPTR_FIFOPTR 112
REG_IOSLAVE_FUPD_FIFOUPD 114
REG_IOSLAVE_FUPD_IOREAD 114
REG_IOSLAVE_IOSCFG_I2CADDR 115
REG_IOSLAVE_IOSCFG_LSB 121
REG_IOSLAVE_PRENC 112
REG_MCU_CTRL_HFRC_HFRCDEL 232
Repeated Count 257
Reset Module 290
RESTART 116
RSTn 290
RTC, *see* real-time clock
RTS 157
S
sampling rate 313
SAR, *see* Successive Approximation Register
SCR, *see* System Control Register
Serial Wire Debugger 35
Single Count 257
Single Pulse 258
Sleep Mode 42, 121
SLEEPONEXIT 42
SLOTs 315
Slots, Mananged Conversion 315
SPHA 95
SPI 92
    3-wire 120
    CHANNEL 93
    CMD Register 92
    Complex Operations 95
    frequency 87
    IO Master 0 3-wire 150
    IO Master 0 3-wire Loopback 153

- IO Master 0 4-wire 147
- IO Master 0 4-wire Loopback 153
- IO Master 1 3-wire 149, 150, 151
- IO Master 1 3-wire Loopback 154
- IO Master 1 4-wire 149
- IO Master 1 4-wire Loopback 153
- IO Slave 3-wire 151
- IO Slave 4-wire 151
- master 86
- Normal Read 93
- Normal Write 93
- OPER 93
- Phase 95
- Polarity 95
- Raw Read 94
- Raw Write 94
- Read 120
- Slave 109, 119
- slave 119
- Slave Addressing 93
- Write 119

SPOL 95, 121
START 116
Start Data Transfer 88, 116
STOP 92, 116
STOP condition 89
Stop Data Transfer 89, 116
Successive Approximation Register 35, 313
SWD, *see* Serial Wire Debugger
SWDCK 159
SWDIO 159
SYSRESETREQn 290
System Control Register 41
T
Temperature Sensor 313
TMRWCR 257, 258
TPIU, *see* Trace Port Interface Unit
Trace Port Interface Unit 35, 43
Track and Hold Time 316
TRIGSEL 315
TX 155
U
UART 155, 230

- CTS 157
- RTS 157
- RX 155

UPLNGTH 92
V
VCOMP, *see* Voltage Comparator
VEXT1 346
Voltage Comparator 159
Voltage Regulator Module 354
VREFEXT1 346
VREFEXT2 346
VREFEXT3 346
VREFINT 346
VTEMP 346
W
Wait-For-Interrupt 42
Wakeup 121
wake-up 35
Wake-Up Interrupt Controller 41
Watchdog Timer 284
WC bit 257
WDTCFG 284
WFI, *see* Wait-For-Interrupt
WIC, *see* wake-up
Window comparator 313
Window Comparisons 316
X
XT 229, 231
XT Oscillator 229

# 17. Ordering Information

**Table 569: Ordering Information**

| Orderable Part Number | Flash | RAM | Package[1,2] | Packing | Temperature Range | Availability |
|---|---|---|---|---|---|---|
| APOLLO512-KCR | 512 KB | 64 KB | 41-pin WLCSP | Tape and Reel | -40 to +85°C | Now |
| APOLLO512-KBR | 512 KB | 64 KB | 64-pin BGA | Tape and Reel | -40 to +85°C | Now |
| APOLLO256-KCR | 256 KB | 32 KB | 41-pin WLCSP | Tape and Reel | -40 to +85°C | Now |
| APOLLO256-KBR | 256 KB | 32 KB | 64-pin BGA | Tape and Reel | -40 to +85°C | Now |

1. For 41-pin WLCSP, minimum order quantity of 5,000 pieces.
   For 64-pin BGA, minimum order quantity of 4,000 pieces.
2. Compliant and certified with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in raw homogeneous materials. The package was designed to be soldered at high temperatures (per reflow profile) and can be used in specified lead-free processes.

# 18. Document Revision History

**Table 570: Document Revision History**

| Revision | Date | Description |
|---|---|---|
| 0.1 | December 2014 | Initial version |
| 0.2 | February 2015 | Initial preliminary version |
| 0.3 | March 2015 | Added register descriptions |
| 0.4 | July 2015 | Added Host Side Address Space and Register section, Ordering Information |
| 0.45 | September 2015 | Footnote c added to Ordering Information, ADC sample time information updated, changed 3.0V spec for current consumption to 3.3V |
| 0.9 | February 2016 | Electrical specifications updated with production values. Several grammatical updates, and details added to peripheral functional descriptions. |
| 1.0 | March 2017 | - Clarification of bit field descriptions throughout<br>- Minor corrections in Pin List<br>- Merged Memory Subsystem into MCU Core Details chapter<br>- Updated IOS functional description<br>- Corrected supply in Pad Connection Details diagram.<br>- Updated Electrical specifications<br>- Footnote about restricted temp range removed from Ordering Information |
| 1.2 | Nov 2023 | ADC:<br>- Updated CFG register’s OPMODE field removing 125 kSps sampling rate option.<br>Electrical:<br>- Updated VESDCDM - ESD Charged Device Model (CDM)<br>- Updated falling slew rate for POR/BOD ($T_{VDDFS}$) in section 15.8 |

## Contact Information

| | |
|---|---|
| **Address** | Ambiq Micro, Inc.<br>6500 River Place Blvd.<br>Building 7, Suite 200<br>Austin, TX 78730-1156 |
| **Phone** | +1 (512) 879-2850 |
| **Website** | https://ambiqmicro.com/ |
| **General Information** | info@ambiqmicro.com |
| **Sales** | sales@ambiqmicro.com |
| **Technical Support** | https://support.ambiqmicro.com |

## Legal Information and Disclaimers

AMBIQ MICRO INTENDS FOR THE CONTENT CONTAINED IN THE DOCUMENT TO BE ACCURATE AND RELIABLE. THIS CONTENT MAY, HOWEVER, CONTAIN TECHNICAL INACCURACIES, TYPOGRAPHICAL ERRORS OR OTHER MISTAKES. AMBIQ MICRO MAY MAKE CORRECTIONS OR OTHER CHANGES TO THIS CONTENT AT ANY TIME. AMBIQ MICRO AND ITS SUPPLIERS RESERVE THE RIGHT TO MAKE CORRECTIONS, MODIFICATIONS, ENHANCEMENTS, IMPROVEMENTS AND OTHER CHANGES TO ITS PRODUCTS, PROGRAMS AND SERVICES AT ANY TIME OR TO DISCONTINUE ANY PRODUCTS, PROGRAMS, OR SERVICES WITHOUT NOTICE.

THE CONTENT IN THIS DOCUMENT IS PROVIDED "AS IS". AMBIQ MICRO AND ITS RESPECTIVE SUPPLIERS MAKE NO REPRESENTATIONS ABOUT THE SUITABILITY OF THIS CONTENT FOR ANY PURPOSE AND DISCLAIM ALL WARRANTIES AND CONDITIONS WITH REGARD TO THIS CONTENT, INCLUDING BUT NOT LIMITED TO, ALL IMPLIED WARRANTIES AND CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHT.

AMBIQ MICRO DOES NOT WARRANT OR REPRESENT THAT ANY LICENSE, EITHER EXPRESS OR IMPLIED, IS GRANTED UNDER ANY PATENT RIGHT, COPYRIGHT, MASK WORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT OF AMBIQ MICRO COVERING OR RELATING TO THIS CONTENT OR ANY COMBINATION, MACHINE, OR PROCESS TO WHICH THIS CONTENT RELATE OR WITH WHICH THIS CONTENT MAY BE USED.

USE OF THE INFORMATION IN THIS DOCUMENT MAY REQUIRE A LICENSE FROM A THIRD PARTY UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF THAT THIRD PARTY, OR A LICENSE FROM AMBIQ MICRO UNDER THE PATENTS OR OTHER INTELLECTUAL PROPERTY OF AMBIQ MICRO.

INFORMATION IN THIS DOCUMENT IS PROVIDED SOLELY TO ENABLE SYSTEM AND SOFTWARE IMPLEMENTERS TO USE AMBIQ MICRO PRODUCTS. THERE ARE NO EXPRESS OR IMPLIED COPYRIGHT LICENSES GRANTED HEREUNDER TO DESIGN OR FABRICATE ANY INTEGRATED CIRCUITS OR INTEGRATED CIRCUITS BASED ON THE INFORMATION IN THIS DOCUMENT. AMBIQ MICRO RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN. AMBIQ MICRO MAKES NO WARRANTY, REPRESENTATION OR GUARANTEE REGARDING THE SUITABILITY OF ITS PRODUCTS FOR ANY PARTICULAR PURPOSE, NOR DOES AMBIQ MICRO ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT, AND SPECIFICALLY DISCLAIMS ANY AND ALL LIABILITY, INCLUDING WITHOUT LIMITATION CONSEQUENTIAL OR INCIDENTAL DAMAGES. “TYPICAL” PARAMETERS WHICH MAY BE PROVIDED IN AMBIQ MICRO DATA SHEETS AND/OR SPECIFICATIONS CAN AND DO VARY IN DIFFERENT APPLICATIONS AND ACTUAL PERFORMANCE MAY VARY OVER TIME. ALL OPERATING PARAMETERS, INCLUDING “TYPICALS” MUST BE VALIDATED FOR EACH CUSTOMER APPLICATION BY CUSTOMER’S TECHNICAL EXPERTS. AMBIQ MICRO DOES NOT CONVEY ANY LICENSE UNDER NEITHER ITS PATENT RIGHTS NOR THE RIGHTS OF OTHERS. AMBIQ MICRO PRODUCTS ARE NOT DESIGNED, INTENDED, OR AUTHORIZED FOR USE AS COMPONENTS IN SYSTEMS INTENDED FOR SURGICAL IMPLANT INTO THE BODY, OR OTHER APPLICATIONS INTENDED TO SUPPORT OR SUSTAIN LIFE, OR FOR ANY OTHER APPLICATION IN WHICH THE FAILURE OF THE AMBIQ MICRO PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR. SHOULD BUYER PURCHASE OR USE AMBIQ MICRO PRODUCTS FOR ANY SUCH UNINTENDED OR UNAUTHORIZED APPLICATION, BUYER SHALL INDEMNIFY AND HOLD AMBIQ MICRO AND ITS OFFICERS, EMPLOYEES, SUBSIDIARIES, AFFILIATES, AND DISTRIBUTORS HARMLESS AGAINST ALL CLAIMS, COSTS, DAMAGES, AND EXPENSES, AND REASONABLE ATTORNEY FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PERSONAL INJURY OR DEATH ASSOCIATED WITH SUCH UNINTENDED OR UNAUTHORIZED USE, EVEN IF SUCH CLAIM ALLEGES THAT AMBIQ MICRO WAS NEGLIGENT REGARDING THE DESIGN OR MANUFACTURE OF THE PART.