

Features

Ultra-low supply current:

- EEMBC ULPBench score of 377
- 35 μ A/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-M4F 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 off-chip sensors:

- 10 bit, 13-channel, up to 800 kSps ADC
- Temperature sensor with $\pm 4^{\circ}\text{C}$ accuracy

Flexible serial peripherals:

- I²C/SPI master for communication with sensors, radios, and other peripherals
- I²C/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 options:

- 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

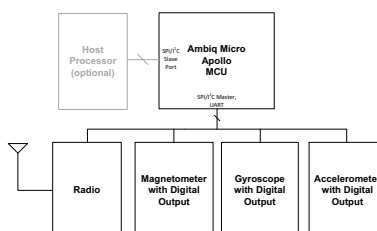
- Wearable electronics
- Wireless sensors
- Activity and fitness monitors
- Consumer electronics
- Consumer medical devices
- Smart watches

Description

The Apollo MCU 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 MCU 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 MCU 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 MCU 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



Contents

1. Package Pins	23
1.1 Pin Configuration	23
1.2 Pin Connections	24
2. System Core	32
3. MCU Core Details	33
3.1 Interrupts And Events	34
3.2 Memory Map	35
3.3 Memory Protection Unit (MPU)	36
3.4 System Busses	37
3.5 Power Management	37
3.5.1 Cortex M4F Power Modes	37
3.6 Debug Interfaces	38
3.6.1 Debugger Attachment	38
3.6.2 Instrumentation Trace Macrocell (ITM)	38
3.6.3 Trace Port Interface Unit (TPIU)	38
3.6.4 Faulting Address Trapping Hardware	39
3.7 ITM Registers	39
3.7.1 Register Memory Map	39
3.7.2 ITM Registers	41
3.8 MCUCTRL Registers	66
3.8.1 Register Memory Map	66
3.8.2 MCUCTRL Registers	67
4. Memory Subsystem	79
4.1 Flash and OTP Memory Subsystem	79
4.1.1 Features	79
4.1.2 Functional Overview	80
4.1.3 Directly Called Flash Helper Functions	80
4.1.4 Debugger Invoked Helper Functions	83
4.1.5 HAL Support for the Flash Helper Functions	85
5. I2C/SPI Master Module	88
5.1 Functional Overview	88
5.2 Interface Clock Generation	88
5.2.1 Command Operation	89
5.2.2 FIFO	89
5.2.3 I2C Interface	90
5.2.4 SPI Operations	94
5.2.5 Repeating a Command	98
5.2.6 Bit Orientation	98
5.2.7 Minimizing Power	98
5.3 IOMSTR Registers	99
5.3.1 Register Memory Map	99

5.3.2 IOMSTR Registers	100
6. I2C/SPI Slave Module	111
6.1 Functional Overview	111
6.2 Local RAM Allocation	111
6.3 Direct Area Functions	112
6.4 FIFO Area Functions	113
6.5 Rearranging the FIFO	114
6.6 Interface Interrupts	114
6.7 Host Address Space and Registers	114
6.8 I2C Interface	114
6.8.1 Bus Not Busy	115
6.8.2 Start Data Transfer	115
6.8.3 Stop Data Transfer	115
6.8.4 Data Valid	115
6.8.5 Acknowledge	115
6.8.6 Address Operation	116
6.8.7 Offset Address Transmission	116
6.8.8 Write Operation	117
6.8.9 Read Operation	117
6.8.10 General Address Detection	118
6.9 SPI Interface	118
6.9.1 Write Operation	118
6.9.2 Read Operation	119
6.9.3 Configuring 3-wire vs. 4-wire SPI Mode	119
6.9.4 SPI Polarity and Phase	119
6.10 Bit Orientation	120
6.11 Wakeup Using the I2C/SPI Slave	120
6.12 IOSLAVE Registers	121
6.12.1 Register Memory Map	121
6.12.2 IOSLAVE Registers	122
6.13 Host Side Address Space and Register	134
6.13.1 Host Address Space and Registers	134
7. GPIO and Pad Configuration Module	139
7.1 Functional Overview	139
7.2 Pad Configuration Functions	139
7.3 General Purpose I/O (GPIO) Functions	142
7.3.1 Configuring the GPIO Functions	142
7.3.2 Reading from a GPIO Pad	142
7.3.3 Writing to a GPIO Pad	142
7.3.4 GPIO Interrupts	142
7.4 Pad Connection Summary	143
7.4.1 Output Selection	143
7.4.2 Output Control	143
7.4.3 Input Control	145

7.4.4 Pull-up Control	145
7.4.5 Analog Pad Configuration	145
7.5 Module-specific Pad Configuration	145
7.5.1 Implementing IO Master Connections	145
7.5.2 Implementing IO Slave Connections	149
7.5.3 Implementing Counter/Timer Connections	152
7.5.4 Implementing UART Connections	153
7.5.5 Implementing GPIO Connections	154
7.5.6 Implementing CLKOUT Connections	155
7.5.7 Implementing ADC Connections	155
7.5.8 Implementing Voltage Comparator Connections	156
7.5.9 Implementing the Software Debug Port Connections	156
7.6 GPIO Registers	157
7.6.1 Register Memory Map	157
7.6.2 GPIO Registers	158
8. Clock Generator and Real Time Clock Module	224
8.1 Clock Generator	224
8.1.1 Functional Overview	224
8.1.2 Low Frequency RC Oscillator (LFRC)	225
8.1.3 High Precision XT Oscillator (XT)	225
8.1.4 High Frequency RC Oscillator (HFRC)	226
8.1.5 HFRC Autoadjustment	226
8.1.6 CLKOUT selection	227
8.1.7 Frequency Measurement	227
8.1.8 Generating 100 Hz	227
8.1.9 XT Oscillator Failure Detection	228
8.1.10 HFRC Stability Delay	228
8.1.11 CLKGEN Registers	229
8.2 Real Time Clock	241
8.2.1 RTC Functional Overview	241
8.2.2 Calendar Counters	241
8.2.3 Calendar Counter Reads	242
8.2.4 Alarms	242
8.2.5 12/24 Hour Mode	243
8.2.6 Century Control and Leap Year Management	243
8.2.7 Weekday Function	243
8.2.8 RTC Registers	243
9. Counter/Timer Module	252
9.1 Functional Overview	252
9.2 Counter/Timer Functions	252
9.2.1 Single Count (FN = 0)	253
9.2.2 Repeated Count (FN = 1)	253
9.2.3 Single Pulse (FN = 2)	254
9.2.4 Repeated Pulse (FN = 3)	254

9.2.5 Continuous (FN = 4)	255
9.3 Creating 32-bit Counters	256
9.4 Power Optimization by Measuring HCLK	256
9.5 Generating the Sample Rate for the ADC	256
9.6 Measuring Buck Converter Charge Insertion	256
9.7 CTIMER Registers	257
9.7.1 Register Memory Map	257
9.7.2 CTIMER Registers	258
10. Watchdog Timer Module	280
10.1 Functional Overview	280
10.2 WDT Registers	281
10.2.1 Register Memory Map	281
10.2.2 WDT Registers	281
11. Reset Generator Module	286
11.1 Functional Overview	286
11.2 External Reset Pin	286
11.3 Power-on Event	287
11.4 Brown-out Event	287
11.5 Software Reset	287
11.6 Watchdog Expiration	287
11.7 RSTGEN Registers	288
11.7.1 Register Memory Map	288
11.7.2 RSTGEN Registers	288
11.7.3 CFG Register	288
12. UART Module	295
12.1 Features	295
12.2 Functional Overview	295
12.3 Enabling and Selecting the UART Clock	296
12.4 Configuration	296
12.5 Transmit FIFO and Receive FIFO	296
12.6 UART Registers	297
12.6.1 Register Memory Map	297
12.6.2 UART Registers	297
13. ADC and Temperature Sensor Module	309
13.1 Features	309
13.2 Functional Overview	310
13.2.1 Clock Source and Dividers	310
13.2.2 13 Channel Analog Mux	310
13.2.3 Triggering and Trigger Sources	310
13.2.4 Voltage Reference Sources	311
13.2.5 Eight Automatically Managed Conversion Slots	311
13.2.6 Automatic Sample Accumulation and Scaling	312
13.2.7 Eight Entry Result FIFO	313

13.2.8 Window Comparator	314
13.3 Operating Modes and the Mode Controller	315
13.3.1 Single Mode	316
13.3.2 Repeat Mode	317
13.3.3 Low Power Modes	317
13.4 Interrupts	317
13.5 Mode Controller State Machine	318
13.6 Voltage Divider and Switchable Battery Load	319
13.7 ADC Registers	320
13.7.1 Register Memory Map	320
13.7.2 ADC Registers	321
14. Voltage Comparator Module	344
14.1 Functional Overview	344
14.2 VCOMP Registers	345
14.2.1 Register Memory Map	345
14.2.2 VCOMP Registers	345
15. Voltage Regulator Module	351
15.1 Functional Overview	351
16. Electrical Characteristics	352
16.1 Absolute Maximum Ratings	352
16.2 Recommended Operating Conditions	353
16.3 Current Consumption	353
16.4 Power Mode Transitions	354
16.5 Clocks/Oscillators	354
16.6 Analog-to-Digital Converter (ADC)	355
16.7 Buck Converter	357
16.8 Power-On RESET (POR) and Brown-Out Detector (BOD)	357
16.9 Resets	358
16.10 Voltage Comparator (VCOMP)	358
16.11 Internal DAC Reference for VCOMP	358
16.12 Inter-Integrated Circuit (I2C) Interface	359
16.13 Serial Peripheral Interface (SPI) Master Interface	360
16.14 Serial Peripheral Interface (SPI) Slave Interface	362
16.15 Universal Asynchronous Receiver/Transmitter (UART)	363
16.16 Counter/Timer (CTIMER)	363
16.17 Flash Memory	364
16.18 General Purpose Input/Output (GPIO)	364
16.19 Serial Wire Debug (SWD)	366
16.20 Serial Wire Output (SWO)	367
16.21 Device Errata and Design Considerations	367
16.21.1 Apollo Device Errata	367
16.21.2 Key Design Guidelines	369
17. Package Mechanical Information	372

17.1 CSP Package	372
17.1.1 PCB land pattern and solder stencil	373
17.2 BGA package	374
17.2.1 PCB land pattern and solder stencil	376
17.3 Reflow Profile	377
18. Ordering Information	383
19. Document Revision History	383

List of Figures

Figure 1. CSP Pin Configuration Diagram (Top View — Balls on Bottom)	23
Figure 2. BGA Pin Configuration Diagram (Top View — Balls on Bottom)	23
Figure 3. Block Diagram for the Apollo Ultra-Low Power MCU	32
Figure 4. ARM Cortex M4F Vector Table	34
Figure 5. Block Diagram for Flash and OTP Subsystem	79
Figure 6. Block diagram for the I2C/SPI Master Module	88
Figure 7. I2C/SPI Master Clock Generation	89
Figure 8. Basic I2C Conditions	90
Figure 9. I2C Acknowledge	91
Figure 10. I2C 7-bit Address Operation	91
Figure 11. I2C 10-bit Address Operation	92
Figure 12. I2C Offset Address Transmission	92
Figure 13. I2C Normal Write Operation	92
Figure 14. I2C Normal Read Operation	93
Figure 15. I2C Raw Write Operation	93
Figure 16. I2C Raw Read Operation	93
Figure 17. SPI Normal Write Operation	95
Figure 18. SPI Normal Read Operation	95
Figure 19. SPI Raw Write Operation	96
Figure 20. SPI Raw Read Operation	96
Figure 21. SPI Combined Operation	97
Figure 22. SPI CPOL and CPHA	97
Figure 23. Block diagram for the I2C/SPI Slave Module	111
Figure 24. I2C/SPI Slave Module LRAM Addressing	112
Figure 25. I2C/SPI Slave Module FIFO	113
Figure 26. Basic I2C Conditions	115
Figure 27. I2C Acknowledge	116
Figure 28. I2C 7-bit Address Operation	116
Figure 29. I2C 10-bit Address Operation	116
Figure 30. I2C Offset Address Transmission	117
Figure 31. I2C Write Operation	117
Figure 32. I2C Read Operation	117
Figure 33. SPI Write Operation	118
Figure 34. SPI Read Operation	119
Figure 35. SPI CPOL and CPHA	119
Figure 36. Block diagram for the General Purpose I/O (GPIO) Module	139
Figure 37. Pad Connection Details	144
Figure 38. Block diagram for the Clock Generator and Real Time Clock Module	224
Figure 39. Block diagram for the Real Time Clock Module	241
Figure 40. Block Diagram for One Counter/Timer Pair	252
Figure 41. Counter/Timer Operation, FN = 0	253
Figure 42. Counter/Timer Operation, FN = 1	254
Figure 43. Counter/Timer Operation, FN = 2	254
Figure 44. Counter/Timer Operation, FN = 3	255

Figure 45. Counter/Timer Operation, FN = 4	256
Figure 46. Block diagram for the Watchdog Timer Module	280
Figure 47. Block diagram for the Reset Generator Module	286
Figure 48. Block diagram of circuitry for Reset pin	287
Figure 49. Block Diagram for the UART Module	295
Figure 50. Block Diagram for ADC and Temperature Sensor	309
Figure 51. Scan Flowchart	316
Figure 52. ADC State Diagram	318
Figure 53. Block diagram for the Voltage Comparator Module	344
Figure 54. Block Diagram for the Voltage Regulator Module	351
Figure 55. I2C Timing	359
Figure 56. SPI Master Mode, Phase = 0	360
Figure 57. SPI Master Mode, Phase = 1	361
Figure 58. SPI Slave Mode, Phase = 0	362
Figure 59. SPI Slave Mode, Phase = 1	363
Figure 60. Serial Wire Debug Timing	366
Figure 61. External Oscillator Example Connection	370
Figure 62. Updated Buck Converter Schematic, Showing C3 Value as 2.0 μ F	371
Figure 63. CSP Package Drawing	372
Figure 64. Example PCB Land Pattern for CSP Package	373
Figure 65. Example Solder Stencil Pattern for CSP Package	373
Figure 66. Example PCB Land Pattern for BGA Package	376
Figure 67. Example Solder Stencil Pattern for BGA Package	376
Figure 68. Reflow Soldering Diagram	377

List of Tables

Table 1: Pin List and Function Table	24
Table 2: ARM Cortex M4F Memory Map	35
Table 3: MCU System Memory Map	35
Table 4: MCU Peripheral Device Memory Map	36
Table 5: ITM Register Map	39
Table 6: STIM0 Register	41
Table 7: STIM0 Register Bits	41
Table 8: STIM1 Register	41
Table 9: STIM1 Register Bits	41
Table 10: STIM2 Register	42
Table 11: STIM2 Register Bits	42
Table 12: STIM3 Register	42
Table 13: STIM3 Register Bits	42
Table 14: STIM4 Register	43
Table 15: STIM4 Register Bits	43
Table 16: STIM5 Register	43
Table 17: STIM5 Register Bits	43
Table 18: STIM6 Register	44
Table 19: STIM6 Register Bits	44
Table 20: STIM7 Register	44
Table 21: STIM7 Register Bits	44
Table 22: STIM8 Register	45
Table 23: STIM8 Register Bits	45
Table 24: STIM9 Register	45
Table 25: STIM9 Register Bits	45
Table 26: STIM10 Register	46
Table 27: STIM10 Register Bits	46
Table 28: STIM11 Register	46
Table 29: STIM11 Register Bits	46
Table 30: STIM12 Register	47
Table 31: STIM12 Register Bits	47
Table 32: STIM13 Register	47
Table 33: STIM13 Register Bits	47
Table 34: STIM14 Register	48
Table 35: STIM14 Register Bits	48
Table 36: STIM15 Register	48
Table 37: STIM15 Register Bits	48
Table 38: STIM16 Register	49
Table 39: STIM16 Register Bits	49
Table 40: STIM17 Register	49
Table 41: STIM17 Register Bits	49
Table 42: STIM18 Register	50
Table 43: STIM18 Register Bits	50
Table 44: STIM19 Register	50

Table 45: STIM19 Register Bits	50
Table 46: STIM20 Register	51
Table 47: STIM20 Register Bits	51
Table 48: STIM21 Register	51
Table 49: STIM21 Register Bits	51
Table 50: STIM22 Register	52
Table 51: STIM22 Register Bits	52
Table 52: STIM23 Register	52
Table 53: STIM23 Register Bits	52
Table 54: STIM24 Register	53
Table 55: STIM24 Register Bits	53
Table 56: STIM25 Register	53
Table 57: STIM25 Register Bits	53
Table 58: STIM26 Register	54
Table 59: STIM26 Register Bits	54
Table 60: STIM27 Register	54
Table 61: STIM27 Register Bits	54
Table 62: STIM28 Register	55
Table 63: STIM28 Register Bits	55
Table 64: STIM29 Register	55
Table 65: STIM29 Register Bits	55
Table 66: STIM30 Register	56
Table 67: STIM30 Register Bits	56
Table 68: STIM31 Register	56
Table 69: STIM31 Register Bits	56
Table 70: TER Register	57
Table 71: TER Register Bits	57
Table 72: TPR Register	57
Table 73: TPR Register Bits	57
Table 74: TCR Register	58
Table 75: TCR Register Bits	58
Table 76: LOCKAREG Register	59
Table 77: LOCKAREG Register Bits	59
Table 78: LOCKSREG Register	59
Table 79: LOCKSREG Register Bits	59
Table 80: PID4 Register	60
Table 81: PID4 Register Bits	60
Table 82: PID5 Register	60
Table 83: PID5 Register Bits	60
Table 84: PID6 Register	61
Table 85: PID6 Register Bits	61
Table 86: PID7 Register	61
Table 87: PID7 Register Bits	61
Table 88: PID0 Register	62
Table 89: PID0 Register Bits	62
Table 90: PID1 Register	62

Table 91: PID1 Register Bits	62
Table 92: PID2 Register	63
Table 93: PID2 Register Bits	63
Table 94: PID3 Register	63
Table 95: PID3 Register Bits	63
Table 96: CID0 Register	64
Table 97: CID0 Register Bits	64
Table 98: CID1 Register	64
Table 99: CID1 Register Bits	64
Table 100: CID2 Register	65
Table 101: CID2 Register Bits	65
Table 102: CID3 Register	65
Table 103: CID3 Register Bits	65
Table 104: MCUCTRL Register Map	66
Table 105: CHIP_INFO Register	67
Table 106: CHIP_INFO Register Bits	67
Table 107: CHIPID0 Register	68
Table 108: CHIPID0 Register Bits	68
Table 109: CHIPID1 Register	68
Table 110: CHIPID1 Register Bits	69
Table 111: CHIPREV Register	69
Table 112: CHIPREV Register Bits	69
Table 113: SUPPLYSRC Register	69
Table 114: SUPPLYSRC Register Bits	70
Table 115: SUPPLYSTATUS Register	70
Table 116: SUPPLYSTATUS Register Bits	70
Table 117: BANDGAPEN Register	71
Table 118: BANDGAPEN Register Bits	71
Table 119: SRAMPWDINSLEEP Register	71
Table 120: SRAMPWDINSLEEP Register Bits	72
Table 121: SRAMPWRDIS Register	73
Table 122: SRAMPWRDIS Register Bits	73
Table 123: FLASHPWRDIS Register	74
Table 124: FLASHPWRDIS Register Bits	74
Table 125: ICODEFAULTADDR Register	75
Table 126: ICODEFAULTADDR Register Bits	75
Table 127: DCODEFAULTADDR Register	75
Table 128: DCODEFAULTADDR Register Bits	75
Table 129: SYSFAULTADDR Register	76
Table 130: SYSFAULTADDR Register Bits	76
Table 131: FAULTSTATUS Register	76
Table 132: FAULTSTATUS Register Bits	76
Table 133: FAULTCAPTUREEN Register	77
Table 134: FAULTCAPTUREEN Register Bits	77
Table 135: TPIUCTRL Register	78
Table 136: TPIUCTRL Register Bits	78

Table 137: IOMSTR Register Map	99
Table 138: FIFO Register	100
Table 139: FIFO Register Bits	100
Table 140: FIFOPTR Register	100
Table 141: FIFOPTR Register Bits	100
Table 142: TLNGTH Register	101
Table 143: TLNGTH Register Bits	101
Table 144: FIFOTHR Register	102
Table 145: FIFOTHR Register Bits	102
Table 146: CLKCFG Register	102
Table 147: CLKCFG Register Bits	102
Table 148: CMD Register	103
Table 149: CMD Register Bits	104
Table 150: CMDRPT Register	104
Table 151: CMDRPT Register Bits	104
Table 152: STATUS Register	105
Table 153: STATUS Register Bits	105
Table 154: CFG Register	106
Table 155: CFG Register Bits	106
Table 156: INTEN Register	107
Table 157: INTEN Register Bits	107
Table 158: INTSTAT Register	108
Table 159: INTSTAT Register Bits	108
Table 160: INTCLR Register	109
Table 161: INTCLR Register Bits	109
Table 162: INTSET Register	110
Table 163: INTSET Register Bits	110
Table 164: IOSLAVE Register Map	121
Table 165: FIFOPTR Register	122
Table 166: FIFOPTR Register Bits	122
Table 167: FIFOCFG Register	122
Table 168: FIFOCFG Register Bits	123
Table 169: FIFOTHR Register	123
Table 170: FIFOTHR Register Bits	123
Table 171: FUPD Register	124
Table 172: FUPD Register Bits	124
Table 173: FIFOCTR Register	124
Table 174: FIFOCTR Register Bits	124
Table 175: FIFOINC Register	125
Table 176: FIFOINC Register Bits	125
Table 177: CFG Register	125
Table 178: CFG Register Bits	126
Table 179: PRENC Register	126
Table 180: PRENC Register Bits	127
Table 181: IOINTCTL Register	127
Table 182: IOINTCTL Register Bits	127

Table 183: GENADD Register	128
Table 184: GENADD Register Bits	128
Table 185: INTEN Register	128
Table 186: INTEN Register Bits	128
Table 187: INTSTAT Register	129
Table 188: INTSTAT Register Bits	129
Table 189: INTCLR Register	130
Table 190: INTCLR Register Bits	130
Table 191: INTSET Register	131
Table 192: INTSET Register Bits	131
Table 193: REGACCINTEN Register	132
Table 194: REGACCINTEN Register Bits	132
Table 195: REGACCINTSTAT Register	132
Table 196: REGACCINTSTAT Register Bits	132
Table 197: REGACCINTCLR Register	133
Table 198: REGACCINTCLR Register Bits	133
Table 199: REGACCINTSET Register	133
Table 200: REGACCINTSET Register Bits	133
Table 201: HOST_IER Register	134
Table 202: HOST_IER Register Bits	134
Table 203: HOST_ISR Register	135
Table 204: HOST_ISR Register Bits	135
Table 205: HOST_WCR Register	135
Table 206: HOST_WCR Register Bits	136
Table 207: HOST_WCS Register	136
Table 208: HOST_WCS Register Bits	136
Table 209: FIFOCTRLO Register	137
Table 210: FIFOCTRLO Register Bits	137
Table 211: FIFOCTRUP Register	137
Table 212: FIFOCTRUP Register Bits	137
Table 213: FIFO Register	138
Table 214: FIFO Register Bits	138
Table 215: Pad Function Mapping	140
Table 216: Pad Function Color and Symbol Code	141
Table 217: Special Pad Types	141
Table 218: I2C Pullup Resistor Selection	142
Table 219: IO Master 0 I2C Configuration	145
Table 220: IO Master 1 I2C Configuration	146
Table 221: IO Master 0 4-wire SPI Configuration	146
Table 223: IO Master 1 4-wire SPI Configuration	147
Table 222: IO Master 0 4-wire SPI nCE Configuration	147
Table 224: IO Master 1 4-wire SPI nCE Configuration	148
Table 225: IO Master 0 3-wire SPI Configuration	148
Table 226: IO Master 1 3-wire SPI Configuration	149
Table 227: IO Slave I2C Configuration	149
Table 228: IO Slave 4-wire SPI Configuration	149

Table 229: IO Slave 3-wire SPI Configuration	150
Table 230: IO Master 0 I2C Loopback	150
Table 231: IO Master 1 I2C Loopback	150
Table 232: IO Master 0 4-wire SPI Loopback	151
Table 233: IO Master 1 4-wire SPI Loopback	151
Table 234: IO Master 0 3-wire SPI Loopback	151
Table 235: IO Master 1 3-wire SPI Loopback	152
Table 236: Counter/Timer Pad Configuration	153
Table 238: UART RX Configuration	154
Table 239: UART RTS Configuration	154
Table 240: UART CTS Configuration	154
Table 237: UART TX Configuration	154
Table 241: CLKOUT Configuration	155
Table 242: ADC Analog Input Configuration	155
Table 244: Voltage Comparator Reference Configuration	156
Table 245: Voltage Comparator Input Configuration	156
Table 243: ADC Trigger Input Configuration	156
Table 246: SWO Configuration	157
Table 247: GPIO Register Map	157
Table 248: PADREGA Register	159
Table 249: PADREGA Register Bits	159
Table 250: PADREGB Register	161
Table 251: PADREGB Register Bits	162
Table 252: PADREGC Register	164
Table 253: PADREGC Register Bits	164
Table 254: PADREGD Register	167
Table 255: PADREGD Register Bits	167
Table 256: PADREGE Register	169
Table 257: PADREGE Register Bits	169
Table 258: PADREGF Register	171
Table 259: PADREGF Register Bits	171
Table 260: PADREGG Register	173
Table 261: PADREGG Register Bits	173
Table 262: PADREGH Register	175
Table 263: PADREGH Register Bits	175
Table 264: PADREGI Register	177
Table 265: PADREGI Register Bits	177
Table 266: PADREGJ Register	179
Table 267: PADREGJ Register Bits	179
Table 268: PADREGK Register	181
Table 269: PADREGK Register Bits	181
Table 270: PADREGL Register	183
Table 271: PADREGL Register Bits	183
Table 272: PADREGM Register	185
Table 273: PADREGM Register Bits	185
Table 274: CFGA Register	186

Table 275: CFGA Register Bits	186
Table 276: CFGB Register	189
Table 277: CFGB Register Bits	189
Table 278: CFGC Register	192
Table 279: CFGC Register Bits	192
Table 280: CFGD Register	194
Table 281: CFGD Register Bits	195
Table 282: CFGE Register	197
Table 283: CFGE Register Bits	197
Table 284: CFGF Register	200
Table 285: CFGF Register Bits	200
Table 286: CFGG Register	202
Table 287: CFGG Register Bits	203
Table 288: PADKEY Register	203
Table 289: PADKEY Register Bits	204
Table 290: RDA Register	204
Table 291: RDA Register Bits	204
Table 292: RDB Register	204
Table 293: RDB Register Bits	205
Table 294: WTA Register	205
Table 295: WTA Register Bits	205
Table 296: WTB Register	205
Table 297: WTB Register Bits	206
Table 298: WTSa Register	206
Table 299: WTSa Register Bits	206
Table 300: WTSB Register	206
Table 301: WTSB Register Bits	207
Table 302: WTCA Register	207
Table 303: WTCA Register Bits	207
Table 304: WTCB Register	207
Table 305: WTCB Register Bits	208
Table 306: ENA Register	208
Table 307: ENA Register Bits	208
Table 308: ENB Register	208
Table 309: ENB Register Bits	209
Table 310: ENSa Register	209
Table 311: ENSa Register Bits	209
Table 312: ENSB Register	209
Table 313: ENSB Register Bits	210
Table 314: ENCA Register	210
Table 315: ENCA Register Bits	210
Table 316: ENCB Register	210
Table 317: ENCB Register Bits	211
Table 318: INT0EN Register	211
Table 319: INT0EN Register Bits	211
Table 320: INT0STAT Register	213

Table 321: INT0STAT Register Bits	213
Table 322: INT0CLR Register	215
Table 323: INT0CLR Register Bits	215
Table 324: INT0SET Register	216
Table 325: INT0SET Register Bits	217
Table 326: INT1EN Register	218
Table 327: INT1EN Register Bits	218
Table 328: INT1STAT Register	220
Table 329: INT1STAT Register Bits	220
Table 330: INT1CLR Register	221
Table 331: INT1CLR Register Bits	221
Table 332: INT1SET Register	222
Table 333: INT1SET Register Bits	222
Table 334: CLKGEN Register Map	229
Table 335: CALXT Register	229
Table 336: CALXT Register Bits	230
Table 337: CALRC Register	230
Table 338: CALRC Register Bits	230
Table 339: ACALCTR Register	230
Table 340: ACALCTR Register Bits	231
Table 341: OCTRL Register	231
Table 342: OCTRL Register Bits	231
Table 343: CLKOUT Register	232
Table 344: CLKOUT Register Bits	232
Table 345: CLKKEY Register	234
Table 346: CLKKEY Register Bits	234
Table 347: CCTRL Register	234
Table 348: CCTRL Register Bits	234
Table 349: STATUS Register	235
Table 350: STATUS Register Bits	235
Table 351: HFADJ Register	236
Table 352: HFADJ Register Bits	236
Table 353: HFVAL Register	237
Table 354: HFVAL Register Bits	237
Table 355: CLOCKS Register	237
Table 356: CLOCKS Register Bits	237
Table 357: UARTEN Register	238
Table 358: UARTEN Register Bits	238
Table 359: INTEN Register	238
Table 360: INTEN Register Bits	238
Table 361: INTSTAT Register	239
Table 362: INTSTAT Register Bits	239
Table 363: INTCLR Register	240
Table 364: INTCLR Register Bits	240
Table 365: INTSET Register	240
Table 366: INTSET Register Bits	241

Table 367: Alarm RPT Function	242
Table 368: RTC Register Map	243
Table 369: CTRLOW Register	244
Table 370: CTRLOW Register Bits	244
Table 371: CTRUP Register	245
Table 372: CTRUP Register Bits	245
Table 373: ALMLOW Register	246
Table 374: ALMLOW Register Bits	246
Table 375: ALMUP Register	247
Table 376: ALMUP Register Bits	247
Table 377: RTCCTL Register	248
Table 378: RTCCTL Register Bits	248
Table 379: INTEN Register	249
Table 380: INTEN Register Bits	249
Table 381: INTSTAT Register	249
Table 382: INTSTAT Register Bits	250
Table 383: INTCLR Register	250
Table 384: INTCLR Register Bits	250
Table 385: INTSET Register	251
Table 386: INTSET Register Bits	251
Table 387: CTIMER Register Map	257
Table 388: TMR0 Register	258
Table 389: TMR0 Register Bits	258
Table 390: CMPRA0 Register	258
Table 391: CMPRA0 Register Bits	258
Table 392: CMPRB0 Register	259
Table 393: CMPRB0 Register Bits	259
Table 394: CTRL0 Register	259
Table 395: CTRL0 Register Bits	260
Table 396: TMR1 Register	262
Table 397: TMR1 Register Bits	262
Table 398: CMPRA1 Register	263
Table 399: CMPRA1 Register Bits	263
Table 400: CMPRB1 Register	263
Table 401: CMPRB1 Register Bits	263
Table 402: CTRL1 Register	264
Table 403: CTRL1 Register Bits	264
Table 404: TMR2 Register	266
Table 405: TMR2 Register Bits	267
Table 406: CMPRA2 Register	267
Table 407: CMPRA2 Register Bits	267
Table 408: CMPRB2 Register	267
Table 409: CMPRB2 Register Bits	268
Table 410: CTRL2 Register	268
Table 411: CTRL2 Register Bits	268
Table 412: TMR3 Register	271

Table 413: TMR3 Register Bits	271
Table 414: CMPRA3 Register	271
Table 415: CMPRA3 Register Bits	271
Table 416: CMPRB3 Register	272
Table 417: CMPRB3 Register Bits	272
Table 418: CTRL3 Register	272
Table 419: CTRL3 Register Bits	273
Table 420: INTEN Register	275
Table 421: INTEN Register Bits	276
Table 422: INTSTAT Register	276
Table 423: INTSTAT Register Bits	276
Table 424: INTCLR Register	277
Table 425: INTCLR Register Bits	277
Table 426: INTSET Register	278
Table 427: INTSET Register Bits	278
Table 428: WDT Register Map	281
Table 429: CFG Register	281
Table 430: CFG Register Bits	281
Table 431: RSTRT Register	282
Table 432: RSTRT Register Bits	282
Table 433: LOCK Register	283
Table 434: LOCK Register Bits	283
Table 435: INTEN Register	283
Table 436: INTEN Register Bits	283
Table 437: INTSTAT Register	284
Table 438: INTSTAT Register Bits	284
Table 439: INTCLR Register	284
Table 440: INTCLR Register Bits	285
Table 441: INTSET Register	285
Table 442: INTSET Register Bits	285
Table 443: RSTGEN Register Map	288
Table 444: CFG Register	288
Table 445: CFG Register Bits	289
Table 446: SWPOI Register	289
Table 447: SWPOI Register Bits	289
Table 448: SWPOR Register	290
Table 449: SWPOR Register Bits	290
Table 450: STAT Register	290
Table 451: STAT Register Bits	290
Table 452: CLRSTAT Register	291
Table 453: CLRSTAT Register Bits	291
Table 454: INTEN Register	292
Table 455: INTEN Register Bits	292
Table 456: INTSTAT Register	292
Table 457: INTSTAT Register Bits	292
Table 458: INTCLR Register	293

Table 459: INTCLR Register Bits	293
Table 460: INTSET Register	293
Table 461: INTSET Register Bits	294
Table 462: UART Register Map	297
Table 463: DR Register	297
Table 464: DR Register Bits	298
Table 465: RSR Register	298
Table 466: RSR Register Bits	298
Table 467: FR Register	299
Table 468: FR Register Bits	299
Table 469: ILPR Register	300
Table 470: ILPR Register Bits	300
Table 471: IBRD Register	301
Table 472: IBRD Register Bits	301
Table 473: FBRD Register	301
Table 474: FBRD Register Bits	301
Table 475: LCRH Register	302
Table 476: LCRH Register Bits	302
Table 477: CR Register	303
Table 478: CR Register Bits	303
Table 479: IFLS Register	304
Table 480: IFLS Register Bits	304
Table 481: IER Register	305
Table 482: IER Register Bits	305
Table 483: IES Register	306
Table 484: IES Register Bits	306
Table 485: MIS Register	307
Table 486: MIS Register Bits	307
Table 487: IEC Register	308
Table 488: IEC Register Bits	308
Table 489: One SLOT Configuration Register	311
Table 490: 10.6 ADC Sample Format	312
Table 491: Per Slot Sample Accumulator	312
Table 492: Accumulator Scaling	313
Table 493: FIFO Register	313
Table 494: Window Comparator Register	314
Table 495: ADC Register Map	320
Table 496: CFG Register	321
Table 497: CFG Register Bits	321
Table 498: STAT Register	323
Table 499: STAT Register Bits	323
Table 500: SWT Register	323
Table 501: SWT Register Bits	324
Table 502: SL0CFG Register	324
Table 503: SL0CFG Register Bits	324
Table 504: SL1CFG Register	326

Table 505: SL1CFG Register Bits	326
Table 506: SL2CFG Register	328
Table 507: SL2CFG Register Bits	328
Table 508: SL3CFG Register	330
Table 509: SL3CFG Register Bits	330
Table 510: SL4CFG Register	332
Table 511: SL4CFG Register Bits	332
Table 512: SL5CFG Register	334
Table 513: SL5CFG Register Bits	334
Table 514: SL6CFG Register	336
Table 515: SL6CFG Register Bits	336
Table 516: SL7CFG Register	337
Table 517: SL7CFG Register Bits	337
Table 518: WLIM Register	339
Table 519: WLIM Register Bits	339
Table 520: FIFO Register	339
Table 521: FIFO Register Bits	340
Table 522: INTEN Register	340
Table 523: INTEN Register Bits	340
Table 524: INTSTAT Register	341
Table 525: INTSTAT Register Bits	341
Table 526: INTCLR Register	342
Table 527: INTCLR Register Bits	342
Table 528: INTSET Register	343
Table 529: INTSET Register Bits	343
Table 530: VCOMP Register Map	345
Table 531: CFG Register	345
Table 532: CFG Register Bits	346
Table 533: STAT Register	347
Table 534: STAT Register Bits	347
Table 535: PWDKEY Register	347
Table 536: PWDKEY Register Bits	348
Table 537: INTEN Register	348
Table 538: INTEN Register Bits	348
Table 539: INTSTAT Register	349
Table 540: INTSTAT Register Bits	349
Table 541: INTCLR Register	349
Table 542: INTCLR Register Bits	350
Table 543: INTSET Register	350
Table 544: INTSET Register Bits	350
Table 545: Absolute Maximum Ratings	352
Table 546: Recommended Operating Conditions	353
Table 547: Current Consumption	353
Table 548: Power Mode Transitions	354
Table 549: Clocks/Oscillators	354
Table 550: Analog to Digital Converter (ADC)	355

Table 551: Buck Converter	357
Table 552: Power-On Reset (POR) and Brown-Out Detector (BOD)	357
Table 553: Resets	358
Table 554: Voltage Comparator (VCOMP)	358
Table 555: Internal DAC Reference for VCOMP	358
Table 556: Inter-Integrated Circuit (I2C) Interface	359
Table 557: Serial Peripheral Interface (SPI) Master Interface	360
Table 558: Serial Peripheral Interface (SPI) Slave Interface	362
Table 559: Universal Asynchronous Receiver/Transmitter (UART)	363
Table 560: Counter/Timer (CTIMER)	363
Table 561: Flash Memory	364
Table 562: General Purpose Input/Output (GPIO)	364
Table 563: Serial Wire Debug (SWD)	366
Table 564: Serial Wire Output (SWO)	367
Table 565: Reflow Soldering Requirements (Pb-free assembly)	377
Table 566: Ordering Information	383
Table 567: Document Revision History	383

1. Package Pins

1.1 Pin Configuration

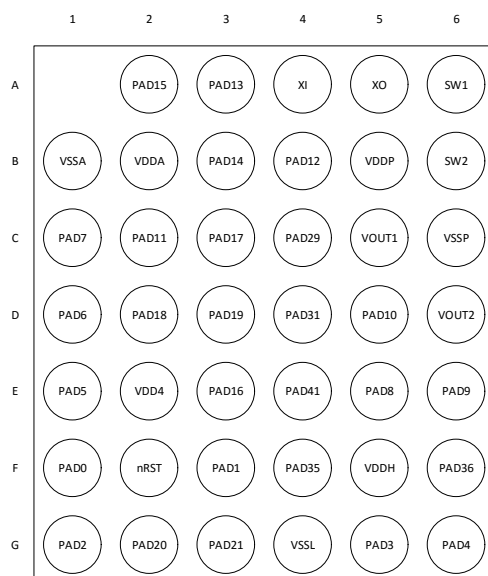


Figure 1. CSP Pin Configuration Diagram (Top View — Balls on Bottom)

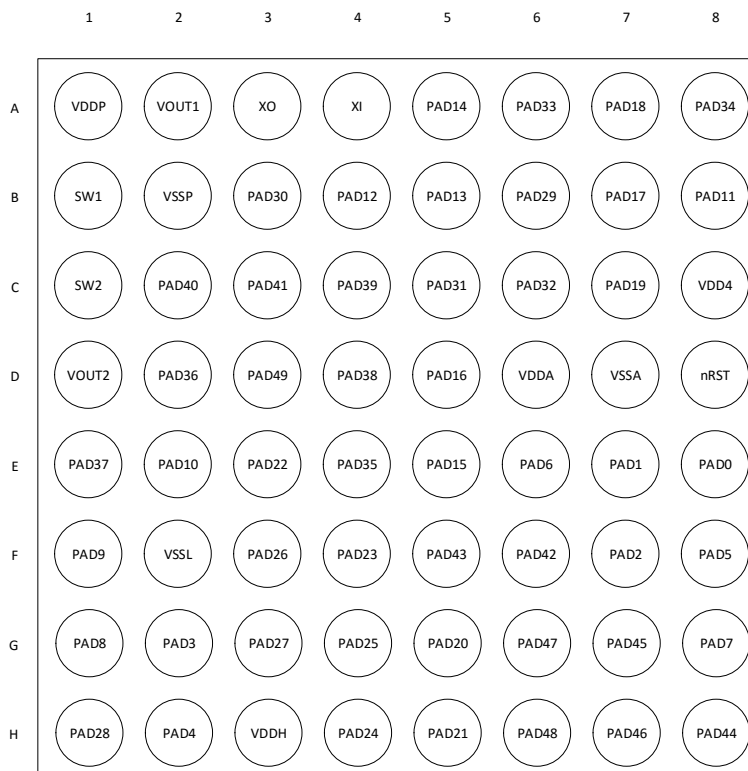


Figure 2. BGA Pin Configuration Diagram (Top View — Balls on Bottom)

1.2 Pin Connections

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	-	-	VOU1	Buck Converter Voltage Output Supply 1	Output
B1	A6	-	-	SW1	Buck Converter 1 Inductor Switch	Power
C1	B6	-	-	SW2	Buck Converter 2 Inductor Switch	Power
D1	D6	-	-	VOU2	Buck Converter Voltage Output Supply 2	Output
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						
A5	B3	14	0	ADC2	Analog to Digital Converter Input 2	Input
			1	M1nCE2	SPI Master 1 Chip Enable 2	Input
			2	UARTTX	UART Transmit Output	Output
			3	GPIO14	General Purpose I/O	Input/Output
A6	-	33	0	CMPRF1	Comparator Reference 1	Input
			1	M0nCE6	SPI Master 0 Chip Enable 6	Input
			3	GPIO33	General Purpose I/O	Input/Output
A7	D2	18	0	CMPIN1	Voltage Comparator Input 1	Input
			1	M0nCE2	SPI Master 0 Chip Enable 2	Input
			2	TCTA1	Timer/Counter A1	Output
			3	GPIO18	General Purpose I/O	Input/Output
A8	-	34	0	CMPRF2	Comparator Reference 2	Input
			1	M0nCE7	SPI Master 0 Chip Enable 7	Input
			3	GPIO34	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
B3	-	30	0	ADC5	Analog to Digital Converter Input 5	Input
			1	M1nCE7	SPI Master 1 Chip Enable 7	Input
			2	TCTB2	Timer/Counter B2	Output
			3	GPIO30	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	Input
			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	Input
			2	TCTB0	Timer/Counter B0	Output
			3	GPIO13	General Purpose I/O	Input/Output
			6	SWO	Serial Wire Debug Output	Output
B6	C4	29	0	ADC4	Analog to Digital Converter Input 4	Input
			1	M1nCE6	SPI Master 1 Chip Enable 6	Input
			2	TCTA2	Timer/Counter A2	Output
			3	GPIO29	General Purpose I/O	Input/Output
B7	C3	17	0	CMPIN0	Voltage Comparator Input 0	Input
			1	M0nCE1	SPI Master 0 Chip Enable 1	Input
			2	TRIG3	ADC Trigger Input 3	Input
			3	GPIO17	General Purpose I/O	Input/Output
B8	C2	11	0	RESERVED		Input
			1	M0nCE0	SPI Master 0 Chip Enable 0	Input
			2	CLKOUT	Oscillator Output Clock	Output
			3	GPIO11	General Purpose I/O	Input/Output
C2	-	40	0	TRIG3	ADC Trigger Input 3	Input
			1	UARTRX	UART Receive Input	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
C4	-	39	0	TRIG2	ADC Trigger Input 2	Input
			1	UARTTX	UART Transmit Output	Output
			2	CLKOUT	Oscillator Output Clock	Output
			3	GPIO39	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
C5	D4	31	0	ADC6	Analog to Digital Converter Input 6	Input
			1	M0nCE4	SPI Master 0 Chip Enable 4	Input
			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	Input
			2	TCTB3	Timer/Counter B3	Output
			3	GPIO32	General Purpose I/O	Input/Output
C7	D3	19	0	CMPRF0	Comparator Reference 0	Input
			1	M0nCE3	SPI Master 0 Chip Enable 3	Input
			2	TCTB1	Timer/Counter B1	Output
			3	GPIO19	General Purpose I/O	Input/Output
D2	F6	36	1	M1nCE1	SPI Master 1 Chip Enable 1	Input
			2	UARTRX	UART Receive Input	Input
			3	GPIO36	General Purpose I/O	Input/Output
D3	-	49	1	M0nCE7	SPI Master 0 Chip Enable 7	Input
			2	TCTB3	Timer/Counter B3	Output
			3	GPIO49	General Purpose I/O	Input/Output
D4	-	38	0	TRIG1	ADC Trigger Input 1	Input
			1	M1nCE3	SPI Master 1 Chip Enable 3	Input
			2	UACTS	UART Clear To Send (CTS) Input	Input
			3	GPIO38	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	Input
			2	TRIG2	ADC Trigger Input 2	Input
			3	GPIO16	General Purpose I/O	Input/Output
E1	-	37	0	TRIG0	ADC Trigger Input 0	Input
			1	M1nCE2	SPI Master 1 Chip Enable 2	Input
			2	UARTS	UART Request To Send (RTS) Output	Output
			3	GPIO37	General Purpose I/O	Input/Output
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	Input
			3	GPIO10	General Purpose I/O	Input/Output
			7	SLWIR3	Loopback SPI Slave I/O Pin for 3-Wire Mode	Bidirectional 3-state

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
E3	-	22	0	UARTTX	UART Transmit Output	Output
			1	M1nCE7	SPI Master 1 Chip Enable 7	Input
			2	TCTA3	Timer/Counter A3	Output
			3	GPIO22	General Purpose I/O	Input/Output
E4	F4	35	1	M1nCE0	SPI Master 1 Chip Enable 0	Input
			2	UARTTX	UART Transmit Output	Output
			3	GPIO35	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	Input
			2	UARTRX	UART Receive Input	Input
			3	GPIO15	General Purpose I/O	Input/Output
E6	D1	6	0	M0SDA	I ² C Master 0 Data	Bidirectional Open Drain
			1	M0MISO	SPI Master 0 Input Data	Input
			2	UACTS	UART Clear To Send (CTS) Input	Input
			3	GPIO6	General Purpose I/O	Input/Output
			4	SLMISO	Loopback SPI Slave Output Data	Output
			6	SLSDA	Loopback I ² C Slave I/O Data	Bidirectional Open Drain
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	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

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
E8	F1	0	0	SLSCL	I ² C Slave Clock	Input
			1	SLSCK	SPI Slave Clock	Input
			2	UARTTX	UART Transmit Output	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	Loopback I ² C Master 1 Clock	Open Drain Output
F1	E6	9	0	M1SDA	I ² C Master 1 Data	Bidirectional Open Drain
			1	M1MISO	SPI Master 1 Input Data	Input
			2	M0nCE5	SPI Master 0 Chip Enable 5	Input
			3	GPIO9	General Purpose I/O	Input/Output
			5	SLMISO	Loopback SPI Slave Output Data	Output
			7	SLSDA	Loopback I ² C Slave Data	Bidirectional Open Drain
F3	-	26	1	M0nCE3	SPI Master 0 Chip Enable 3	Input
			2	TCTB0	Timer/Counter B0	Output
			3	GPIO26	General Purpose I/O	Input/Output
F4	-	23	0	UARTRX	UART Receive Input	Input
			1	M0nCE0	SPI Master 0 Chip Enable 0	Input
			2	TCTB3	Timer/Counter B3	Output
			3	GPIO23	General Purpose I/O	Input/Output
F5	-	43	0	TRIG6	ADC Trigger Input 6	Input
			1	M0nCE1	SPI Master 0 Chip Enable 1	Input
			2	TCTB0	Timer/Counter B0	Output
			3	GPIO43	General Purpose I/O	Input/Output
F6	-	42	0	TRIG5	ADC Trigger Input 5	Input
			1	M0nCE0	SPI Master 0 Chip Enable 0	Input
			2	TCTA0	Timer/Counter A0	Output
			3	GPIO42	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
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
F8	E1	5	0	M0SCL	I ² C Master 0 Clock	Open Drain Output
			1	M0SCK	SPI Master 0 Clock	Output
			2	UARTS	UART Request To Send (RTS) Output	Output
			3	GPIO5	General Purpose I/O	Input/Output
			4	M0SCK	Loopback SPI Master 0 Clock	Output
			6	M0SCL	Loopback I ² C Master 0 Clock	Open Drain Output
G1	E5	8	0	M1SCL	I ² C Master 1 Clock	Open Drain Output
			1	M1SCK	SPI Master 1 Clock	Output
			2	M0nCE4	SPI Master 0 Chip Enable 4	Input
			3	GPIO8	General Purpose I/O	Input/Output
			5	M1SCK	Loopback SPI Master 1 Clock	Output
			7	M1SCL	Loopback I ² C Master 1 Clock	Open Drain Output
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	Input
			3	GPIO3	General Purpose I/O	Input/Output
			4	M0nCE	Loopback SPI Master 0 Chip Enable	Input
			5	M1nCE	Loopback SPI Master 1 Chip Enable	Input
G3	-	27	1	M1nCE4	SPI Master 1 Chip Enable 4	Input
			2	TCTA1	Timer/Counter A1	Output
			3	GPIO27	General Purpose I/O	Input/Output
G4	-	25	1	M0nCE2	SPI Master 0 Chip Enable 2	Input
			2	TCTA0	Timer/Counter A0	Output
			3	GPIO25	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
G5	G2	20	0	SWDCK	Software Debug Clock	Input
			1	M1nCE5	SPI Master 1 Chip Enable 5	Input
			2	TCTA2	Timer/Counter A2	Output
			3	GPIO20	General Purpose I/O	Input/Output
G6	-	47	1	M0nCE5	SPI Master 0 Chip Enable 5	Input
			2	TCTB2	Timer/Counter B2	Output
			3	GPIO47	General Purpose I/O	Input/Output
G7	-	45	1	M0nCE3	SPI Master 0 Chip Enable 3	Input
			2	TCTB1	Timer/Counter B1	Output
			3	GPIO45	General Purpose I/O	Input/Output
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
H1	-	28	1	M1nCE5	SPI Master 1 Chip Enable 5	Input
			2	TCTB1	Timer/Counter B1	Output
			3	GPIO28	General Purpose I/O	Input/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	Input
			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
H4	-	24	1	M0nCE1	SPI Master 0 Chip Enable 1	Input
			2	CLKOUT	Oscillator Output Clock	Output
			3	GPIO24	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	Input
			2	TCTB2	Timer/Counter B2	Output
			3	GPIO21	General Purpose I/O	Input/Output
H6	-	48	1	M0nCE6	SPI Master 0 Chip Enable 6	Input
			2	TCTA3	Timer/Counter A3	Output
			3	GPIO48	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
H7	-	46	1	M0nCE4	SPI Master 0 Chip Enable 4	Input
			2	TCTA2	Timer/Counter A2	Output
			3	GPIO46	General Purpose I/O	Input/Output
H8	--	44	0	TRIG7	ADC Trigger Input 7	Input
			1	M0nCE2	SPI Master 0 Chip Enable 2	Input
			2	TCTA1	Timer/Counter A1	Output
			3	GPIO44	General Purpose I/O	Input/Output

2. System Core

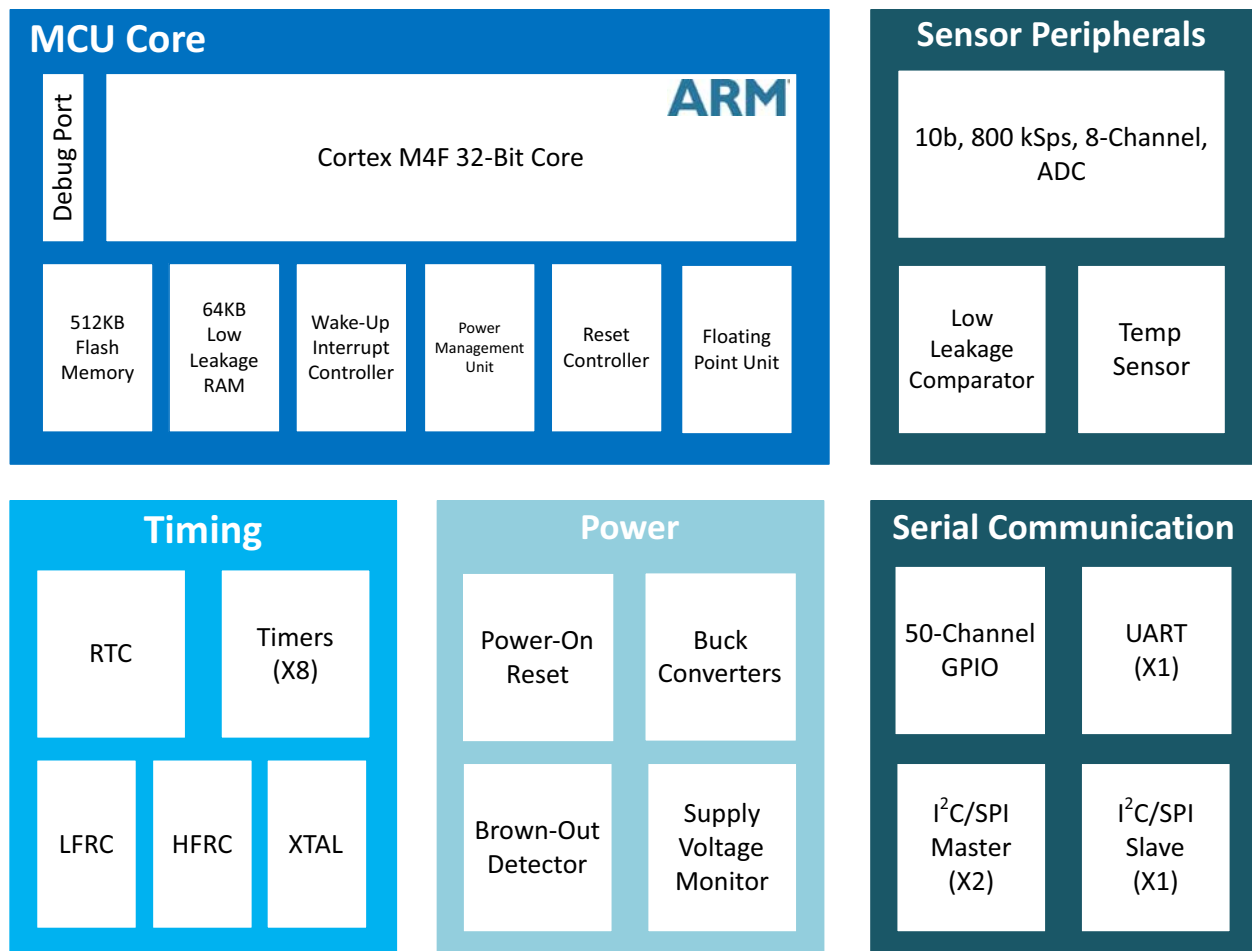


Figure 3. Block Diagram for the Apollo Ultra-Low Power MCU

The Apollo Ultra-Low Power MCU, shown in Figure 3, is an ideal solution for battery-powered applications requiring sensor measurement and data analysis. In a typical system, the Apollo MCU serves as an applications processor for one or more sensors and/or radios. The Apollo MCU can measure analog sensor outputs using an integrated ultra-low power 10 bit ADC and digital sensor outputs using the integrated serial master ports. The Cortex-M4F core integrated in the Apollo MCU family is capable of running complex data analysis and sensor fusion algorithms to process the sensor data. The Cortex-M4F core 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 Apollo MCU over its serial slave port using the I²C or SPI protocol.

With unprecedented energy efficiency for sensor conversion and data analysis, the Apollo MCU 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 MCU enables the use of more complex sensor processing algorithms due to its extremely low active mode power of <40 μ A/MHz. By using the Apollo MCU, 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 MCU is a 32-bit ARM Cortex M4F processor with several tightly coupled peripherals. The Ambiq Micro implementation of the Cortex M4F 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 RAM. The Wake-Up Interrupt Controller (WIC) coupled with the Cortex M4F 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 $\pm 2^{\circ}\text{C}$ 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 I²C/SPI master port and/or UART port enables the MCU to communicate with external sensors and radios (such as Bluetooth transceivers) that have digital outputs.

The Apollo Ultra-Low Power MCU 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 MCU 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 MCU 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

At the center of the Apollo MCU is a 32-bit ARM Cortex M4F 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 M4F 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 I²C/SPI master modules), base memory addresses for the registers are provided for each and noted as INSTANCE 0, INSTANCE 1, etc.

3.1 Interrupts And Events

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 M4F 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 M4F core when it is in a deep sleep (SRPG) mode.

The MCU supports the M4F NMI as well as the normal interrupt types. For details on the Interrupt model of the M4F, please see the “**Cortex-M4F Devices Generic User Guide**,” document number DUI0553A.

Below is the M4F Vector Table:

Exception Number	IRQ Number	Offset	Vector	Peripheral/Description
255	239	0x03FC	IRQ239	
.	.	.	.	
.	.	.	.	
.	.	0x0070		
.	.	0x006C	IRQ11	UART
.	.	0x0068	IRQ10	Counter/Timers
.	.	0x0064	IRQ9	GPIO
.	.	0x0060	IRQ8	ADC
.	.	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				
9				
8				
7				
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	

Figure 4. ARM Cortex M4F Vector Table

The Cortex-M4F 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 M4F 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.

3.2 Memory Map

ARM has a well-defined memory map for devices based on the ARM v7-M Architecture. The M4F 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 2: ARM Cortex M4F 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 – 0xE0FFFFFF	Private Peripheral Bus	N	NVIC, System timers, System Control Block
0xE0100000 – 0xFFFFFFFF	Vendor	N	Vendor Defined

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

Table 3: 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 – 0x0800FFFF	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 – 0xE0FFFFFF	PPB	N	NVIC, System timers, System Control Block
0xE0100000 – 0xEFFFFFFF	Reserved	X	No device at this address range
0xF0000000 – 0xF000FFFF	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 4 shows the address mapping for the peripheral devices of the Base Platform.

Table 4: 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

3.3 Memory Protection Unit (MPU)

The Apollo MCU includes an MPU which is a core component for memory protection. The M4F 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.

3.4 System Busses

The ARM Cortex-M4F 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 MCU 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 MCU 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.

3.5 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 M4F 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 M4F is then returned to active mode once all state is ready.

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

3.5.1 Cortex M4F Power Modes

The ARM Cortex M4F defines the following 3 power modes:

- Active
- Sleep
- Deep Sleep

Each mode is described below.

3.5.1.1 Active Mode

In the Active Mode, the M4F is powered up, clocks are active, and instructions are being executed. In this mode, the M4F 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 M4F 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 M4F 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

3.5.1.2 Sleep Mode

In the Sleep Mode, the M4F is powered up, but the clocks (HCLK, FCLK) are not active. The power supply is still applied to the M4F logic such that it can immediately become active on a wakeup event and begin executing instructions.

3.5.1.3 Deep Sleep Mode

In the Deep Sleep Mode, the M4F 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 M4F will handshake with the Wake-up Interrupt Controller and Power Management Unit and set up the possible wakeup conditions.

3.6 Debug Interfaces

A number of useful debug facilities are provided in the Apollo MCU.

3.6.1 Debugger Attachment

An external debugger can be connected to the MCU using ARM's 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.

3.6.2 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.

3.6.3 Trace Port Interface Unit (TPIU)

The Apollo MCU includes a Cortex-M4F Trace Port Interface Unit (TPIU). The Cortex-M4F 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.

3.6.4 Faulting Address Trapping Hardware

The Apollo MCU 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.

3.7 ITM Registers

ARM ITM Registers

INSTANCE 0 BASE ADDRESS:0x00000000

3.7.1 Register Memory Map

Table 5: 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

Table 5: ITM Register Map

Address(s)	Register Name	Description
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
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

3.7.2.29 STIM28 Register

Stimulus Port Register 28

OFFSET: 0xE0000070

INSTANCE 0 ADDRESS: 0xE0000070

Stimulus Port Register 28

Table 62: STIM28 Register

3	3	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	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
STIM28																															

Table 63: STIM28 Register Bits

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

3.7.2.30 STIM29 Register

Stimulus Port Register 29

OFFSET: 0xE0000074

INSTANCE 0 ADDRESS: 0xE0000074

Stimulus Port Register 29

Table 64: STIM29 Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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 65: STIM29 Register Bits

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

3.7.2.35 TCR Register

Trace Control Register.

OFFSET: 0xE0000E80

INSTANCE 0 ADDRESS: 0xE0000E80

Trace Control Register

Table 74: TCR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD								BUSY	ATB_ID								RSVD				TS_FREQ	TS_PRESCALE	RSVD				SWV_ENABLE	DWT_ENABLE	SYNC_ENABLE	TS_ENABLE	ITM_ENABLE

Table 75: 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.
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.

3.7.2.36 LOCKAREG Register

Lock Access Register

OFFSET: 0xE000FB0**INSTANCE 0 ADDRESS: 0xE000FB0**

Lock Access Register

Table 76: LOCKAREG Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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
LOCKAREG																															

Table 77: LOCKAREG Register Bits

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

3.7.2.37 LOCKSREG Register

Lock Status Register

OFFSET: 0xE0000FB4

INSTANCE 0 ADDRESS: 0xE000FB4

Lock Status Register

Table 78: LOCKSREG Register

[illegible]

Table 79: 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 MCUCTRL Registers

MCU Miscellaneous Control Logic

INSTANCE 0 BASE ADDRESS:0x40020000

3.8.1 Register Memory Map

Table 104: 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.

3.8.2 MCUCTRL Registers

3.8.2.1 CHIP_INFO Register

Chip Information Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x40020000

Chip Information Register

Table 105: CHIP_INFO Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
CLASS								FLASH				RAM				MAJORREV				MINORREV				PKG		PINS		TEMP	QUAL		

Table 106: CHIP_INFO Register Bits

Bit	Name	Reset	RW	Description
31:24	CLASS	0x1	RO	Device class. APOLLO = 0x1 - APOLLO
23:20	FLASH	0x4	RO	Device flash size. 256K = 0x3 - 256K of available flash. 512K = 0x4 - 512K of available flash.
19:16	RAM	0x1	RO	Device RAM size. 32K = 0x0 - 32K of available SRAM. 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. BGA = 0x2 - Ball grid array. CSP = 0x3 - Chip-scale package.
5:3	PINS	0x1	RO	Number of pins. 41PINS = 0x1 - 41 package pins total. 64PINS = 0x1 - 64 package pins total.
2:1	TEMP	0x1	RO	Device temperature range. COMMERCIAL = 0x0 - Commercial temperature range.

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

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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
VALUE																															

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

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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
VALUE																															

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

[illegible]

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

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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
RSVD																												COREBUCKEN	MEMBUCKEN		

Table 114: 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. 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. EN = 0x1 - Enable the Memory Buck as the supply for flash and SRAM.

3.8.2.6 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 115: SUPPLYSTATUS Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																													COREBUCKON	MEMBUCKON	

Table 116: 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. LDO = 0x0 - Indicates the the LDO is supplying the Core low-voltage. 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. LDO = 0x0 - Indicates the LDO is supplying the memory power domain. BUCK = 0x1 - Indicates the Buck is supplying the memory power domain.

3.8.2.7 BANDGAPEN Register

Band Gap Enable

OFFSET: 0x000000FC

INSTANCE 0 ADDRESS: 0x400200FC

Band Gap Enable

Table 117: BANDGAPEN Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
RSVD																																BGPEN

Table 118: BANDGAPEN Register Bits

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

3.8.2.8 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 119: SRAMPWDINSLEEP Register

3	3	2	2	2	2	2	2	2	2	2	2	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
RSVD																						BANK7	BANK6	BANK5	BANK4	BANK3	BANK2	BANK1	BANK0		

Table 120: 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. NORMAL = 0x0 - SRAM Bank 7 normal operation. 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. NORMAL = 0x0 - SRAM Bank 6 normal operation. 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. NORMAL = 0x0 - SRAM Bank 5 normal operation. 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. NORMAL = 0x0 - SRAM Bank 4 normal operation. 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. NORMAL = 0x0 - SRAM Bank 3 normal operation. 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. NORMAL = 0x0 - SRAM Bank 2 normal operation. 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. NORMAL = 0x0 - SRAM Bank 1 normal operation. 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. NORMAL = 0x0 - SRAM Bank 0 normal operation. PWRDN_IN_DEEPSLEEP = 0x1 - SRAM Bank 0 deep sleep.

3.8.2.9 SRAMPWRDIS Register

Disables individual banks of the SRAM array

OFFSET: 0x00000144

INSTANCE 0 ADDRESS: 0x40020144

Disables individual banks of the SRAM array

Table 121: SRAMPWRDIS Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																								BANK7	BANK6	BANK5	BANK4	BANK3	BANK2	BANK1	BANK0

Table 122: 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. 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. 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. 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. 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. 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. 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. DIS = 0x1 - Disable SRAM Bank 1.

Table 122: 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. DIS = 0x1 - Disable SRAM Bank 0.

3.8.2.10 FLASHPWRDIS Register

Disables individual banks of the Flash array

OFFSET: 0x00000148

INSTANCE 0 ADDRESS: 0x40020148

Disables individual banks of the Flash array

Table 123: FLASHPWRDIS Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																												BANK1	BANK0		

Table 124: 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. 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. DIS = 0x1 - Disable Flash instance 0.

3.8.2.13 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 129: SYSFAULTADDR Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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
ADDR																															

Table 130: 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.

3.8.2.14 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 131: FAULTSTATUS Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																												SYS	DCODE	ICODE	

Table 132: FAULTSTATUS Register Bits

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

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. NOFAULT = 0x0 - No bus fault has been detected. 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. NOFAULT = 0x0 - No DCODE fault has been detected. 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. NOFAULT = 0x0 - No ICODE fault has been detected. FAULT = 0x1 - ICODE fault detected.

3	3	2	2	2	2	2	2	2	2	2	2	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	
RSVD																																ENABLE

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. DIS = 0x0 - Disable fault capture. EN = 0x1 - Enable fault capture.

3.8.2.16 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 135: TPIUCTRL Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																					CLKSEL	RSVD							ENABLE		

Table 136: 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. LOW_PWR = 0x0 - Low power state. 0MHZ = 0x0 - Low power state. 6MHZ = 0x1 - Selects 6MHz frequency. 3MHZ = 0x2 - Selects 3MHz frequency. 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. DIS = 0x0 - Disable the TPIU. EN = 0x1 - Enable the TPIU.

4. Memory Subsystem

4.1 Flash and OTP Memory Subsystem

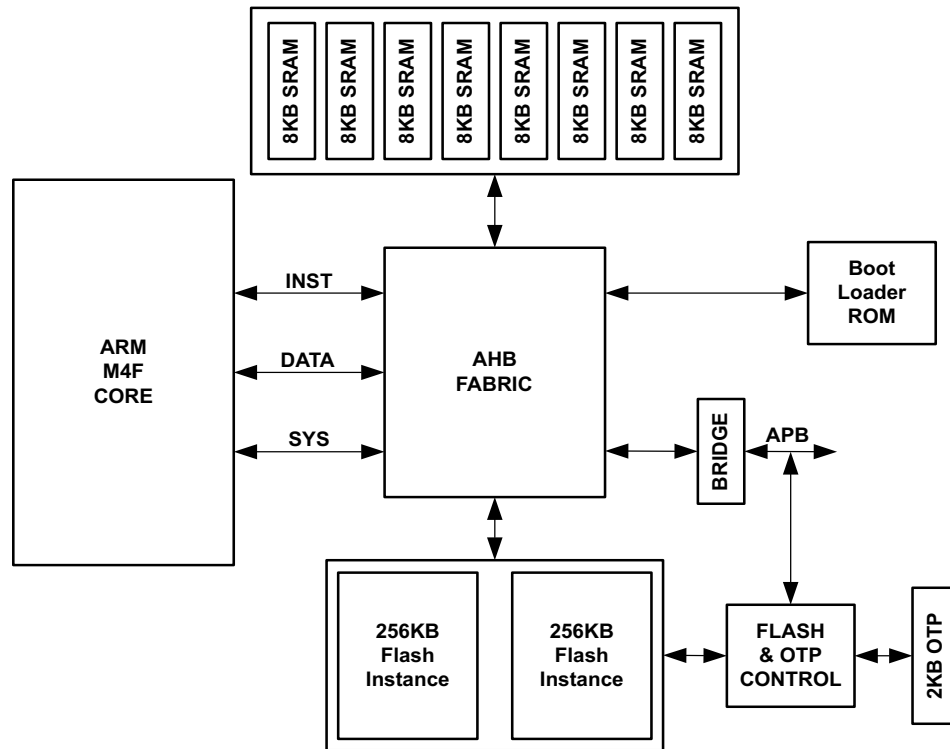


Figure 5. Block Diagram for Flash and OTP Subsystem

4.1.1 Features

The Apollo MCU integrates four kinds of memory as shown in Figure 5:

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

This chapter discusses the latter three types.

Key features include:

- 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

4.1.2 Functional Overview

The Apollo MCU Integrates up to 512 KBytes of on-board flash memory and 2048 bytes of one time programmable memory. These two 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 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 contains approximately 2 KB of 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.

Flash helper functions can be called directly from code executing in flash to erase or program other pages of flash memory than the one in execution. The helper functions include a mass erase function that will erase the entire contents of a flash instance. Obviously this type of function must be called from code running in SRAM or there will be no caller function to return to. Finally, the helper functions include operations that can be invoked directly from a debugger. For these functions, a few parameters are initialized in SRAM and a body of code is copied to SRAM. The debugger sets the PC to the flash helper function entry point and resumes the M4F processor from that point. This type of helper function hits a hard coded breakpoint at the end of execution and returns to the debugger. The latter are used to initially download programs into flash.

4.1.3 Directly Called Flash Helper Functions

The following flash helper functions are intended to be called directly from programs running either in flash or in SRAM. These are standard C functions that perform the standard operations on either the flash or the OTP.

4.1.3.1 Directly Called Function to Erase Main Block of Flash

Use this function to erase the entire contents of one instance (256 KB) of a flash.

```
/******//**
```

```
* @brief entry point for erasing the MAIN block of one entire flash instance
```

```
*
```

```
* This function operates on one flash instance per call.
```

```
*
```

```
* Calling this function erases an entire flash instance (256 KB).
```

```
*
```

```
* @param value PROGRAM key
```

```
* @param flash_inst 0 or 1 selects first or second flash block
```

```
* @return 0 for success, non-zero for failure
```

```
*
```

```
*****/
```

```
#define PROGRAM_KEY    (0x12344321)

int flash_mass_erase(
    uint32_t value,
    uint32_t flash_inst
);
```

4.1.3.2 *Directly Called Function to Erase one Page in the Flash*

One page in the flash is 2048 bytes. This function can be used to erase one page at a time.

```
/**
 * @brief entry point for erasing one page in the MAIN block of one flash instance
 *
 * This function operates on one page of the MAIN block of one flash instance per call.
 *
 * Calling this function erases one page in the MAIN block of one flash instance (2048 bytes).
 *
 * @param value PROGRAM key
 * @param flash_inst 0 or 1 selects first or second flash block
 * @param PageNumber offset to a page aligned target location to begin page erasing in the flash MAIN block
 * @return 0 for success, non-zero for failure
 */
#define PROGRAM_KEY    (0x12344321)

int flash_page_erase(
    uint32_t value,
    uint32_t flash_inst,
    uint32_t PageNumber
);
```

4.1.3.3 *Directly Called Function to Program N Words in the User Portion of the OTP*

Use this function to write words in the OTP such as the protection bits or other user desired permanent information such as a unique ID.

```
/**
 * @brief entry point for programming the user portion (1024 bytes) of the 2048 byte OTP.
 *
 * Calling this function programs multiple words in the OTP
 *
 * @param value      OTP program key
 * @param OTP_inst    always set to zero
 * @param pSrc        pointer to word aligned array of data to program into the OTP.
 * @param Offset      word offset in to the OTP (0x01 means the second 32 bit word)
 *
 * NOTE Offset == 0 --> first word in second half of OTP, 0x400 into the 2048 byte page
 */
```

```
* @param NumberOfWords number of words to program
*
* NOTE Only the upper half of the OTP can be written so 1 < NumberOfWords < 257
* @return 0 for success, non-zero for failure
*
```

```
*****/
```

```
#define OTP_PROGRAM_KEY (0x87655678)

int flash_program_otp_area(
    uint32_t value,
    uint32_t OTP_inst,
    uint32_t *pSrc,
    uint32_t Offset,    // 0 <= Offset < 256
    uint32_t NumberOfWords // 1 < NumberOfWords < 257
);
```

4.1.3.4 Directly Called Function to Program N words in the Flash

Use this function to write words to an erased block of flash. NOTE: the flash does not support overwriting words without an intervening erase operation.

```
/******//**
```

```
* @brief Program N words of the MAIN block of one flash instance
*
* Calling this function programs N words of the MAIN block of 1 flash instance
*
* @param value PROGRAM key
* @param flash_inst 0 or 1 selects first or second flash block.
* @param pSrc pointer to word aligned data to program into the flash MAIN block.
*
* WARNING: pSrc can not point back in to flash itself
* @param pDst pointer to word aligned target location to begin programming in the flash MAIN block
*
* NOTE: we discard any bits above 256 K and below bit 2 (stays within one instance & word aligned)
* @param NumberOfWords number of 32-bit words to program
* @return 0 for success, non-zero for failure
*
*
```

```
*****/
```

```
#define PROGRAM_KEY (0x12344321)

int flash_program_main(
    uint32_t value,
    uint32_t *pSrc,
    uint32_t *pDst,
    uint32_t NumberOfWords
);
```

4.1.4 *Debugger Invoked Helper Functions*

The following functions are intended to be invoked from an external debugger or debugger surrogate such as a manufacturing gang programmer. The debugger sets up a few parameters in SRAM and copies the binary data to be programmed in to the flash in to the SRAM behind the parameters. It then sets the PC to the entry point for this function and resumes execution from there. When the function completes, it hits a hard coded break point instruction which passes control back to the debugger or debugger surrogate.

4.1.4.1 *Program Words In The Flash From a Debugger*

The debugger uses this function to copy binary data from SRAM to a target location in flash.

```

/*****
* @brief entry point for reading program details and data from SRAM and then programming it into MAIN
*
* Calling this function looks up programming information from offset 0x0 in SRAM
*   0x10000000 pointer in to flash
*   0x10000004 number of 32-bit words to program
*   0x10000008 PROGRAM key to pass to flash helper routine
*   0x1000000C return code debugger sets this to -1 all RCs are >= 0
*   0x10000010 first 32-bit word of data buffer to be programmed
*
* Note this routine hits a break point instruction to transfer control to a debugger surrogate in the parallel programmer.
*
* @return never returns, spins here waiting for debugger or debugger surrogate on the parallel programmer
*
*****/
#define PROGRAM_KEY    (0x12344321)
void flash_program_main_from_sram(void);

```

4.1.4.2 *Program Words in the OTP From a Debugger*

The debugger uses this function to copy binary data from SRAM to a target location in the OTP.

```

/*****
* @brief entry point for reading program details and data from SRAM and then programming it into the OTP
*
* Calling this function looks up programming information from offset 0x0 in SRAM
*   0x10000000 Offset in to the OTP block (user portion)
*       0 <= Offset < 256
*       256 added to offset before programming
*   0x10000004 number of 32-bit words to program
*   0x10000008 OTP program key to pass to flash helper routine
*   0x1000000C return code debugger sets this to -1 all RCs are >= 0
*   0x10000010 first 32-bit word of data buffer to be programmed
*
*****/

```

* Note this routine hits a break point instruction to transfer control to a debugger surrogate in the parallel programmer.

*

* @return never returns, spins here waiting for debugger or debugger surrogate on the parallel programmer

*

*

*****/

```
#define OTP_PROGRAM_KEY (0x87655678)
```

```
void flash_program_otp_area_from_sram(void);
```

4.1.4.3 Erase a Number of Contiguous Pages From a Debugger

The debugger uses this function to erase a number of contiguous pages in flash.

/******//**

* @brief entry point for erasing a contiguous block of main block pages from SRAM

*

* Calling this function looks up page erase information from offset 0x0 in SRAM

* 0x10000000 instance number

* 0x10000004 number of main block pages to erase must be between 1 and 128 inclusive

* 0 < number < 129

* 0x10000008 PROGRAM key to pass to flash helper routine

* 0x1000000C return code debugger sets this to -1 all RCs are >= 0

* 0x10000010 PageNumber of the first flash page to erase. NOTE: these now *HAVE* to be sequential

* range 0 <= PageNumber <= 127

*

* Note this routine hits a break point instruction to transfer control to a debugger surrogate in the parallel programmer.

*

* @return never returns, spins here waiting for debugger or debugger surrogate on the parallel programmer

*

*

*****/

```
void flash_erase_main_pages_from_sram(void);
```

4.1.4.4 Erase an Entire Flash Instance From a Debugger

The debugger uses this function to an entire flash instance.

```

/*****
* @brief entry point for reading mass erase parameters from SRAM and then erasing the main block
*
* Calling this function looks up programming information from offset 0x0 in SRAM
*      0x10000000 pointer in to flash instance #
*      0x10000004 program key to pass to flash helper routine
*      0x10000008 return code debugger sets this to -1 all RCs are >= 0
*
* Note this routine spins when flash_mass_erase() returns and waits for the debugger surrogate in the parallel
programmer.
*
* @return never returns, spins here waiting for debugger or debugger surrogate on the parallel programmer
*
*****/
void flash_mass_erase_from_sram(void);

```

4.1.5 HAL Support for the Flash Helper Functions

The AmbiqSuite firmware Hardware Abstraction Layer (HAL) provides routines for finding and accessing all of the directly called flash helper functions. The HAL contains a function pointer table to gain access to the flash helper functions which looks like this:

```

//
// Structure of function pointers to helper functions for invoking various
// flash operations.
//
typedef struct am_hal_flash_helper_struct{
    int (*am_hal_flash_mass_erase)(uint32_t, uint32_t);
    int (*am_hal_flash_page_erase)(uint32_t, uint32_t, uint32_t);
    int (*am_hal_flash_program_main)(uint32_t, uint32_t *, uint32_t *, uint32_t);
    int (*am_hal_flash_program_otp)(uint32_t, uint32_t, uint32_t *, uint32_t, uint32_t);
    void (*am_hal_flash_program_main_sram)(void);
    void (*am_hal_flash_program_otp_sram)(void);
    void (*am_hal_flash_erase_main_pages_sram)(void);
    void (*am_hal_flash_mass_erase_sram)(void);
} g_am_hal_flash_t;
extern g_am_hal_flash_t g_am_hal_flash;

```

It also contains constant address assignment pointers like this:

```
//  
// Look-up table  
//  
g_am_hal_flash_t g_am_hal_flash = {  
    // am_hal_flash_mass_erase  
    ((int (*)(uint32_t, uint32_t)) 0x0800004d),  
    // am_hal_flash_page_erase  
    ((int (*)(uint32_t, uint32_t, uint32_t)) 0x08000051),  
    // am_hal_flash_program_main  
    ((int (*)(uint32_t, uint32_t *, uint32_t *, uint32_t)) 0x08000055),  
    // am_hal_flash_program_otp  
    ((int (*)(uint32_t, uint32_t, uint32_t *, uint32_t, uint32_t)) 0x08000059),  
    // am_hal_flash_program_main_sram  
    ((void (*)(void)) 0x0800005d),  
    // am_hal_flash_program_otp_sram  
    ((void (*)(void)) 0x08000061),  
    // am_hal_flash_erase_main_pages_sram  
    ((void (*)(void)) 0x08000065),  
    // am_hal_flash_mass_erase_sram  
    ((void (*)(void)) 0x08000069)  
};
```

Finally, as an example of the HAL abstraction of the flash helper functions:

```
/*******  
//  
//! @brief This programs up to N bytes of the Main array on one flash Instance.  
//!  
//! @param ui32Value - The Program key.  
//! @param pui32Src - Pointer to word aligned array of data to program into  
//! the flash block.  
//! @param pui32Dst - Pointer to word aligned location to begin programming  
//! the flash block.  
//! @param ui32NumWords - The Number of words to program.  
//!  
//! This function will program multiple words in the FLAASH instance  
//! instance 0.  
//!
```



```
/*!
/*! @return 0 for success, non-zero for failure.
//
//*****
int
am_hal_flash_program_main(uint32_t ui32Value, uint32_t *pui32Src,
                          uint32_t *pui32Dst, uint32_t ui32NumWords)
{
    return g_am_hal_flash.am_hal_flash_program_main(ui32Value, pui32Src,
                                                    pui32Dst, ui32NumWords);
}
```

5. I²C/SPI Master Module

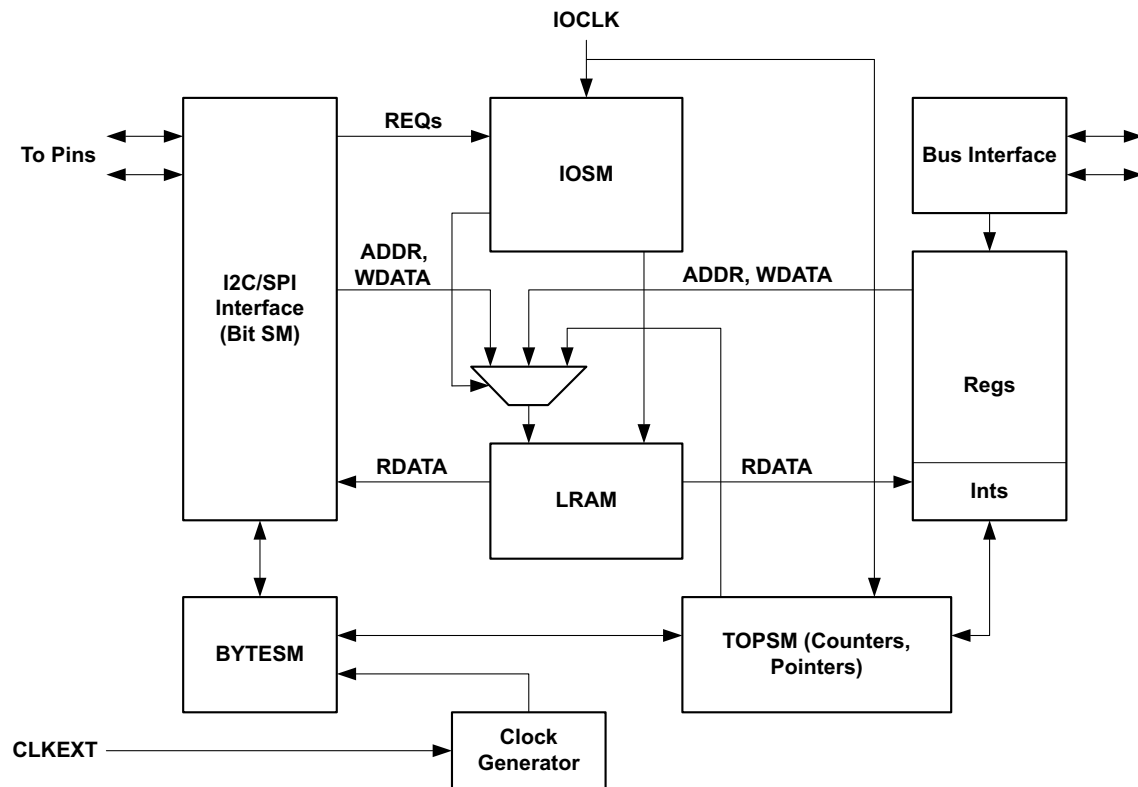


Figure 6. Block diagram for the I²C/SPI Master Module

5.1 Functional Overview

The Apollo MCU includes two I²C/SPI Master Modules, shown in Figure 6, each of which functions as the Master of an I²C or SPI interface as selected by the REG_IOMSTRn_IOMCFG_IFCSEL bit (n=0 or 1). A 64-byte bidirectional FIFO and a sophisticated Command mechanism allow simple initiation of I/O operations without requiring software interaction.

In I²C mode the I²C/SPI Master supports 7- and 10-bit addressing, multi-master arbitration, interface frequencies from 1.2 kHz to 1.0 MHz and up to 255-byte burst operations. In SPI mode the I²C/SPI Master supports up to 8 slaves with automatic nCE selection, 3 and 4-wire implementation, interface frequencies from 1.2 kHz to 8 MHz, all SPI polarity/phase combinations and up to 4095-byte burst operations, with both standard embedded address operations and raw read/write transfers.

5.2 Interface Clock Generation

The I²C/SPI Master can generate a wide range of I/O interface clocks, as shown in Figure 7. The source clock is a scaled version of the HFRC 24 MHz clock, selected by REG_IOMSTRn_CLKCFG_FSEL. A divide-by-3 circuit may be selected by REG_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, REG_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 REG_IOMSTRn_CLKCFG_DIVEN, the length of the low period of the clock is specified by REG_IOMSTRn_CLKCFG_LOWPER + 1. Otherwise, the clock will have a 50% duty cycle.

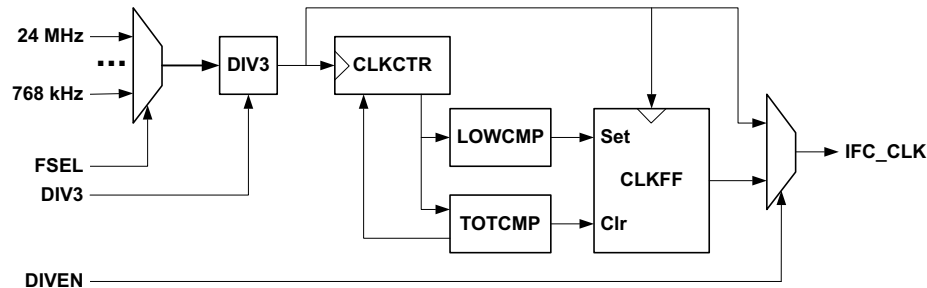


Figure 7. I²C/SPI Master Clock Generation

5.2.1 Command Operation

In order to minimize the amount of time the CPU must be awake during I²C/SPI Master operations, the architecture of the I²C/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 REG_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 I²C/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 REG_IOMSTRn_FIFOTHR_FIFOWTHR so the CPU can wake up and refill the FIFO. The I²C/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 REG_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 I²C/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.

5.2.2 FIFO

The I²C/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 REG_IOMSTRn_FIFOPTR_FIFOSIZ register by 4 and decrement the REG_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, FIFORTH and FIFOWTHR indicate when a THR interrupt should be generated to signal the processor that data should be transferred.

5.2.3 I²C Interface

The I²C/SPI Master supports a flexible set of Commands to implement a variety of standard I²C operations. The I²C 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 MCU I²C 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 8) and are described in the following sections.

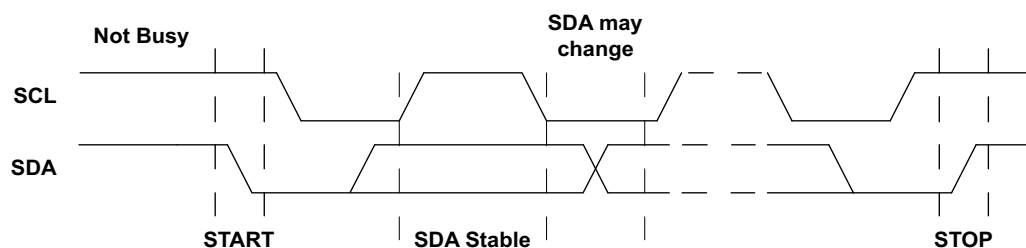


Figure 8. Basic I²C Conditions

5.2.3.1 Bus Not Busy

Both SDA and SCL remain high.

5.2.3.2 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.

5.2.3.3 Stop Data Transfer

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

5.2.3.4 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.

5.2.3.5 Acknowledge

Each byte of eight bits is followed by one acknowledge (ACK) bit as shown in Figure 9. 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.

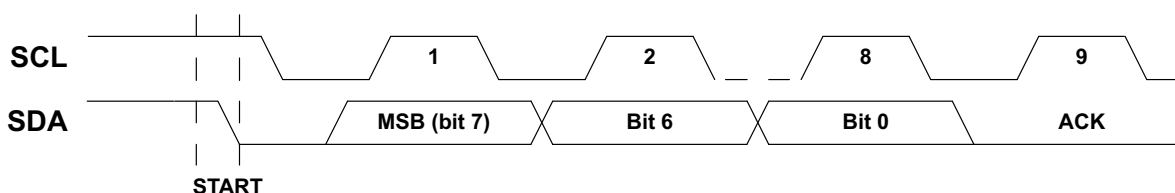


Figure 9. I²C Acknowledge

5.2.3.6 I²C Slave Addressing

For normal I²C 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 10 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.

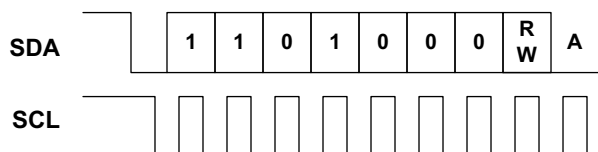


Figure 10. I²C 7-bit Address Operation

Figure 11 shows the operation with which the master addresses the Apollo MCU 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.

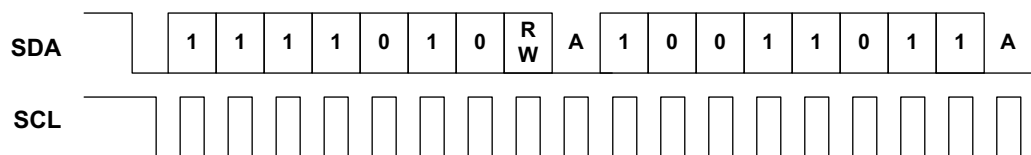


Figure 11. I²C 10-bit Address Operation

5.2.3.7 I²C Offset Address Transmission

If the OPER field of the CMD selects a Normal Read or Write, the I²C/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 12. The Offset Address is loaded into the Address Pointer of the slave.

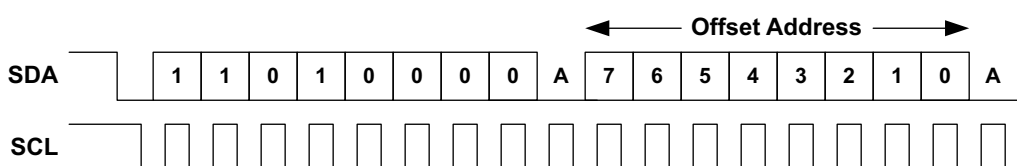


Figure 12. I²C Offset Address Transmission

5.2.3.8 I²C Normal Write Operation

In a Normal write operation the I²C/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 12. 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 13.

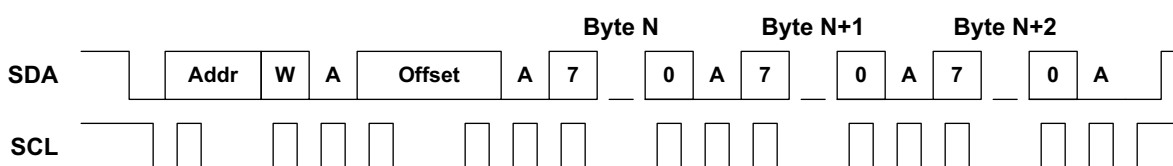


Figure 13. I²C Normal Write Operation

5.2.3.9 I²C Normal Read Operation

If a Normal Read operation is selected in the OPER field of the Command, the I²C/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 14, 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 I²C/SPI Master receiver responds with a NAK and a STOP to complete the operation.

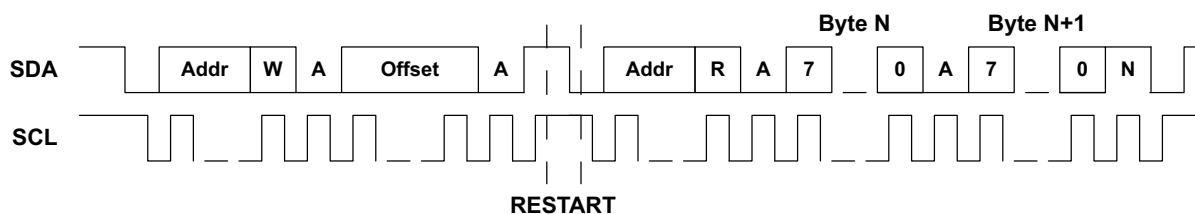


Figure 14. I²C Normal Read Operation

5.2.3.10 I²C Raw Write Operation

If a Raw Write is selected in the OPER field of the Command, the I²C/SPI Master does not execute the Offset Address Transmission, but simply begins transferring bytes as shown in Figure 15. This provides support for slave devices which do not implement the standard offset address architecture. The OFFSET field of Command is not used in this case.

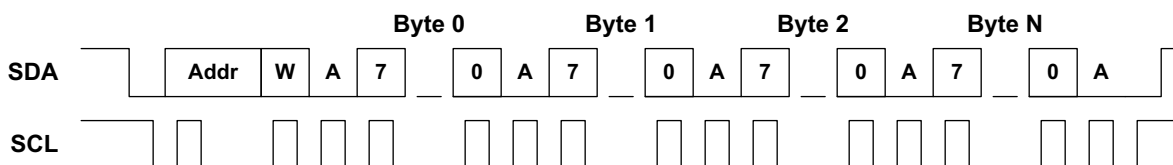


Figure 15. I²C Raw Write Operation

5.2.3.11 I²C Raw Read Operation

If a Raw Read is selected in the OPER field of Command, the I²C/SPI Master does not execute the Offset Address Transmission, but simply begins transferring bytes with a read as shown in Figure 16. 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.

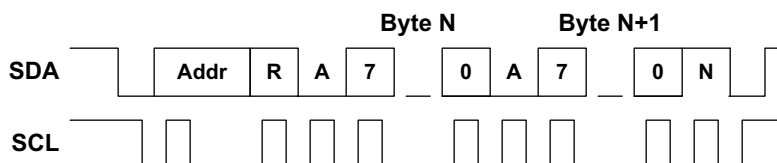


Figure 16. I²C Raw Read Operation

5.2.3.12 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 MCU to hold the interface between Commands to insure that another master does not inadvertently access the same slave that the Apollo MCU 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 if CONT is set the read transaction will not terminate with a STOP.

5.2.3.13 I²C Multi-master Arbitration

The Apollo MCU 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 MCU 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 MCU. 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 MCU 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.

5.2.4 SPI Operations

5.2.4.1 SPI Configuration

The I²C/SPI Master supports all combinations of the polarity (CPOL) and phase (CPHA) modes of SPI using the REG_IOMSTRn_IOMCFG_SPOL and REG_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 provide a 12-bit length specification.

5.2.4.2 SPI Slave Addressing

In SPI mode, the Command specifies the slave channel to be used in the CHANNEL field. The I²C/SPI Master supports up to eight 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.

5.2.4.3 SPI Normal Write

Figure 17 shows the case of a SPI Normal Write operation selected in the OPER field. The operation is initiated when the I²C/SPI Master pulls one of the eight nCE signals low. At that point the I²C/SPI Master begins generating the clock on SCK and 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. The entire Offset Address Byte, including the RW 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 RW bit. If the slave does not expect a RW 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 I²C/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 I²C/SPI Master.

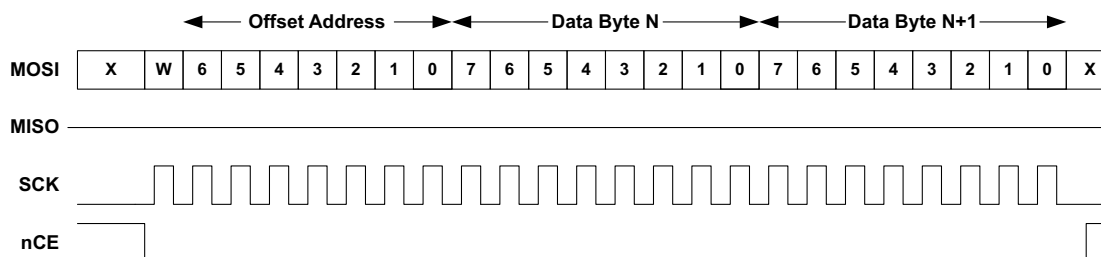


Figure 17. SPI Normal Write Operation

5.2.4.4 SPI Normal Read

Figure 18 shows the case of a Normal Read operation selected in the OPER field. The operation is initiated when the I²C/SPI Master pulls one of the eight nCE signals low. At that point the I²C/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 RW bit is a 0 indicating a read. After the transfer of the last address bit (bit 0), the I²C/SPI Master stops driving the MOSI line and begins loading the FIFO with the data on the MISO line. The transfer continues until the I²C/SPI Master brings the nCE line high.

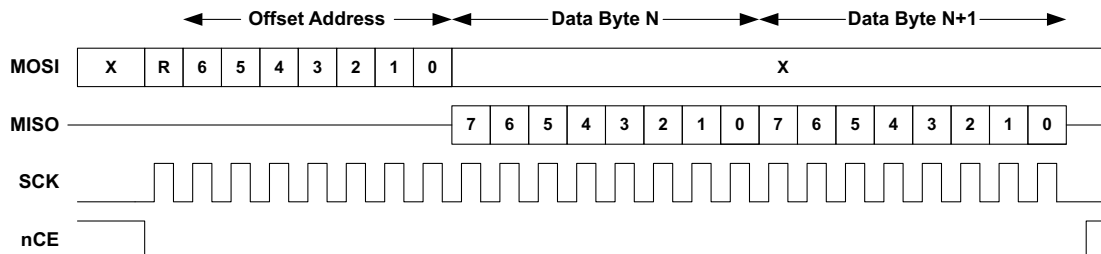


Figure 18. SPI Normal Read Operation

As with a Normal Write, the Offset Address byte including the RW bit is taken from the OFFSET field of CMD. If the slave expects an RW bit, OFFSET[7] must be set to 1. This allows reads from devices which have different formats for the address byte.

5.2.4.5 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 19. The OFFSET field is not used in this case.

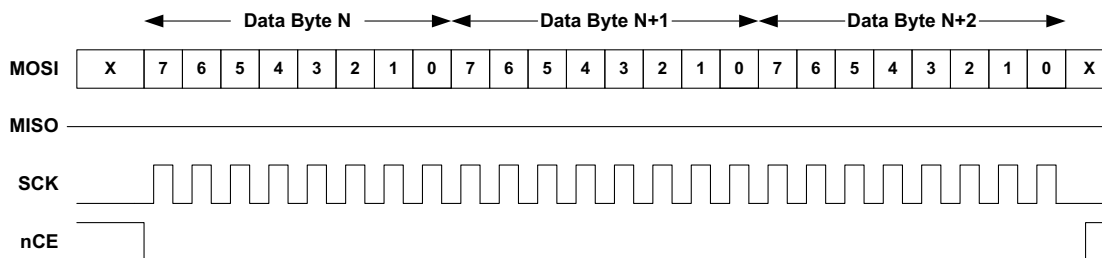


Figure 19. SPI Raw Write Operation

5.2.4.6 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 20. The OFFSET field is not used in this case.

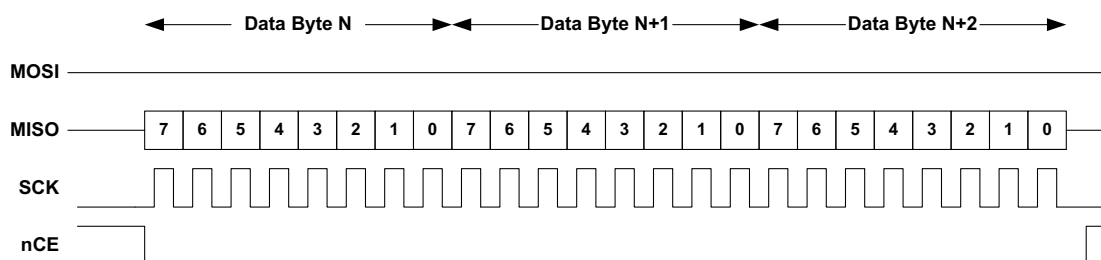


Figure 20. SPI Raw Read Operation

5.2.4.7 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 21 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.

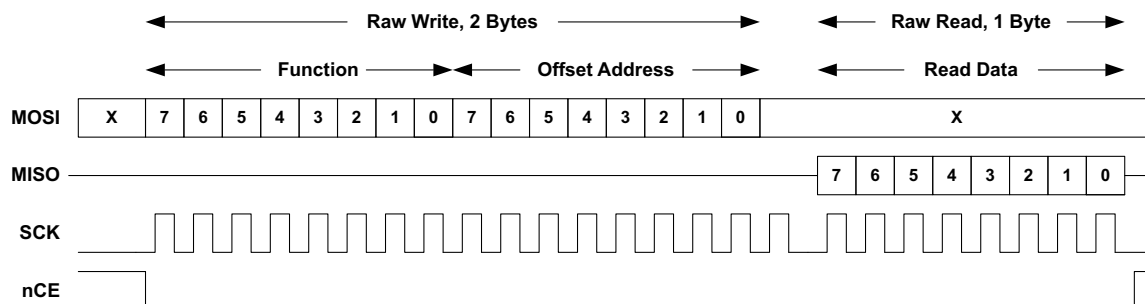


Figure 21. SPI Combined Operation

5.2.4.8 SPI Polarity and Phase

The Apollo MCU supports all combinations of CPOL (clock polarity) and CPHA (data phase) in SPI mode, as defined by the SPOL and SPHA bits. Figure 22 shows how these two bits affect the interface signal behavior.

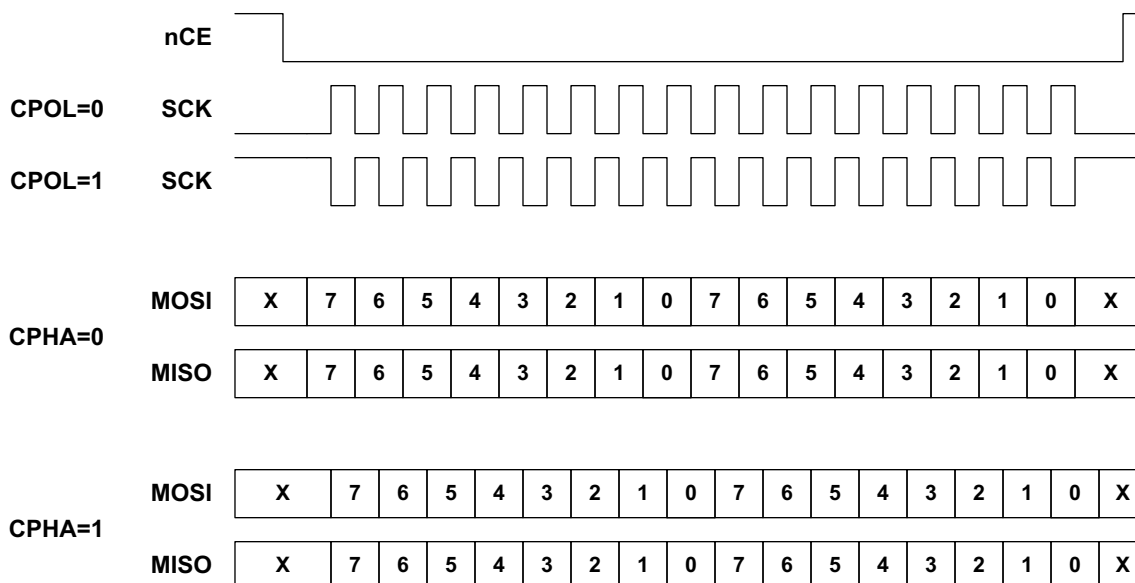


Figure 22. SPI CPOL and CPHA

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.

5.2.5 Repeating a Command

Some peripherals, particularly sensors such as accelerometers and gyroscopes, have multiple registers which hold sample data (2 bytes each of X, Y and Z are common), and FIFOs behind these registers which hold multiple samples. In order to allow software to retrieve several samples with a single operation, the Apollo MCU I²C/SPI Master includes the capability to execute the same Command multiple times. If multiple Commands are desired, the REG_IOMSTRn_CMDRPT Register is loaded with the number of additional times to execute the next Command (e.g. loading CMDRPT with the value 3 will cause the next Command to be executed a total of 4 times). When a Command is written to the Command Register, the Command is then executed multiple times, filling or emptying the FIFO as appropriate. The series of repeated Commands behaves as if it was a single long Command, with a single CMDCMP interrupt occurring at the end and THR interrupts occurring if the FIFO crosses the relevant threshold. At the end of any Command the CMDRPT Register has the value 0, so that single Commands may always be executed without requiring a write to CMDRPT.

As an example, assume the peripheral has 6 bytes of sensor sample data located at register offsets 10, 11, 12, 13, 14 and 15. Also assume the internal FIFO threshold of the peripheral has been set so that an interrupt occurs when the FIFO contains 8 samples. The CMDRPT register is set to 7, and a read Command is executed with an offset of 10 and a length of 6. This Command will be executed 8 times, each time bursting 6 bytes of data from registers 10-15 in the peripheral to the I²C/SPI Master FIFO. When CMDCMP is received the FIFO in the I²C/SPI Master will contain 48 bytes of data. The bytes of data are packed in the FIFO – there are no gaps between samples.

5.2.6 Bit Orientation

In both I²C and SPI modes, the I²C/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.

5.2.7 Minimizing Power

Each I²C/SPI Master has a global interface enable bit REG_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 a 0.

5.3 IOMSTR Registers

I2C/SPI Master

INSTANCE 0 BASE ADDRESS:0x50004000

INSTANCE 1 BASE ADDRESS:0x50005000

5.3.1 Register Memory Map

Table 137: IOMSTR Register Map

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

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.

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	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						
RSVD																				TLNGTH																	

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

Table 144: FIFOTHR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																		FIFOWTHR					RSVD	FIFORTH							

Table 145: 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	FIFORTH	0x0	RW	FIFO read threshold.

5.3.2.5 CLKCFG Register

I/O Clock Configuration

OFFSET: 0x0000010C

INSTANCE 0 ADDRESS: 0x5000410C

INSTANCE 1 ADDRESS: 0x5000510C

I/O Clock Configuration

Table 146: CLKCFG Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
TOTPER								LOWPER								RSVD		DIVEN	DIV3	FSEL		RSVD									

Table 147: 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.

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

3	3	2	2	2	2	2	2	2	2	2	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
CMD																															

Table 149: CMD Register Bits

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

5.3.2.7 CMDRPT Register

Command Repeat Register

OFFSET: 0x00000114**INSTANCE 0 ADDRESS: 0x50004114**

INSTANCE 1 ADDRESS: 0x50005114

Command Repeat Register

Table 150: CMDRPT Register

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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	0						
RSVD																									CMDRPT													

Table 151: 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.

5.3.2.9 CFG Register

I/O Master Configuration

OFFSET: 0x0000011C

INSTANCE 0 ADDRESS: 0x5000411C

INSTANCE 1 ADDRESS: 0x5000511C

I/O Master Configuration

Table 154: CFG Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
IFCEN	RSVD																												SPHA	SPOL	IFCSEL

Table 155: CFG Register Bits

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

5.3.2.12 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 160: INTCLR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																					ARB	STOP	START	ICMD	IACC	WTLEN	NAK	FOVFL	FUNDFL	THR	CMDCMP

Table 161: 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.

5.3.2.13 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 162: INTSET Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																					ARB	STOP	START	ICMD	IACC	WTLEN	NAK	FOVFL	FUNDFL	THR	CMDCMP

Table 163: 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.

6. I²C/SPI Slave Module

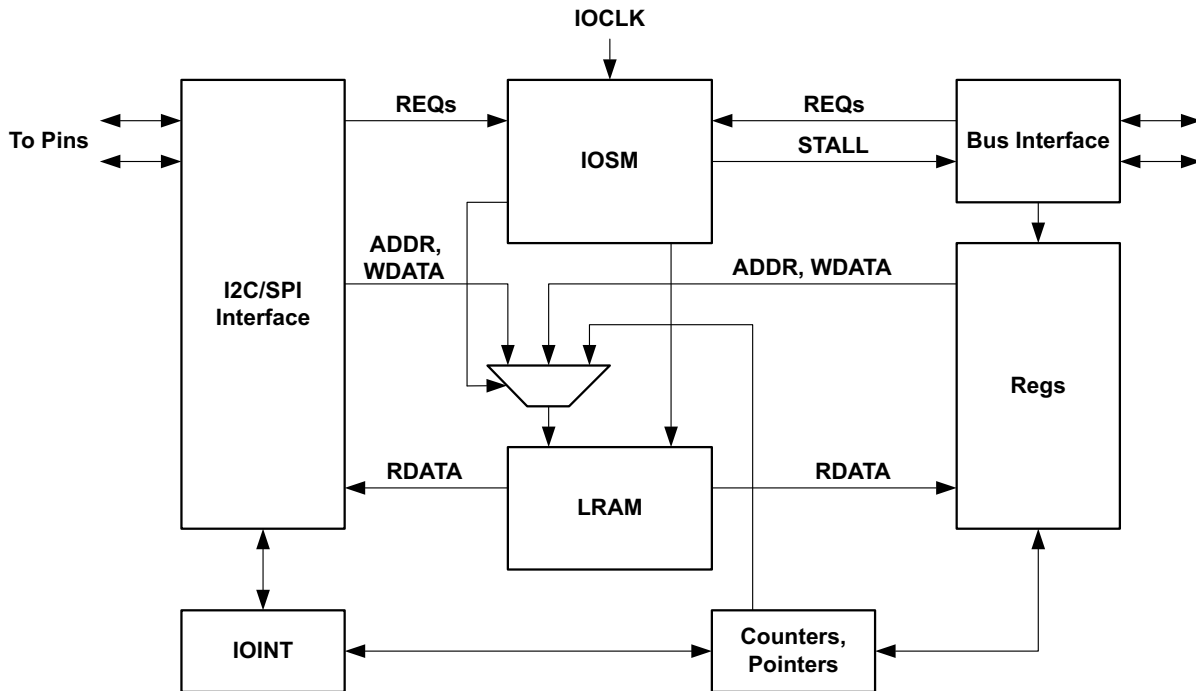


Figure 23. Block diagram for the I²C/SPI Slave Module

6.1 Functional Overview

The I²C/SPI Slave Module, shown in Figure 23, allows the Apollo MCU to function as a Slave in an I²C or SPI system. The I²C/SPI Slave operates in an independent fashion, so that the Apollo MCU may be placed into 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 I²C/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 I²C mode the Slave supports fully configurable 7 and 10-bit addressing and interface frequencies up to 1.2 MHz. In SPI mode, the Slave supports all polarity/phase combinations and interface frequencies up to 10 MHz.

6.2 Local RAM Allocation

The I²C/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 I²C/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 which is mapped between the AHB address space and the I/O address space. This area is from LRAM address 0x00 to the address in the 5-bit configuration register REG_IOSLAVE_FIFOCFG_FIFOBASE minus 1. FIFOBASE should be at most 0xF (120 bytes of directly mapped locations).
2. A FIFO Area which is used to stream data from the Apollo MCU. 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 in the FIFOBASE register to the LRAM address in the REG_IOSLAVE_FIFOCFG_FIFOMAX register minus 1.
3. A RAM Area which is accessible only from the AHB Slave. The RAM area is from the LRAM address in the FIFOMAX register to address 0xFF. The data in the LRAM is maintained in Deep Sleep Mode.

Figure 24 below shows the LRAM address mapping between the I/O interface and the AHB. Note that FIFOMAX may be set to 0x20, in which case there will be no RAM Area.

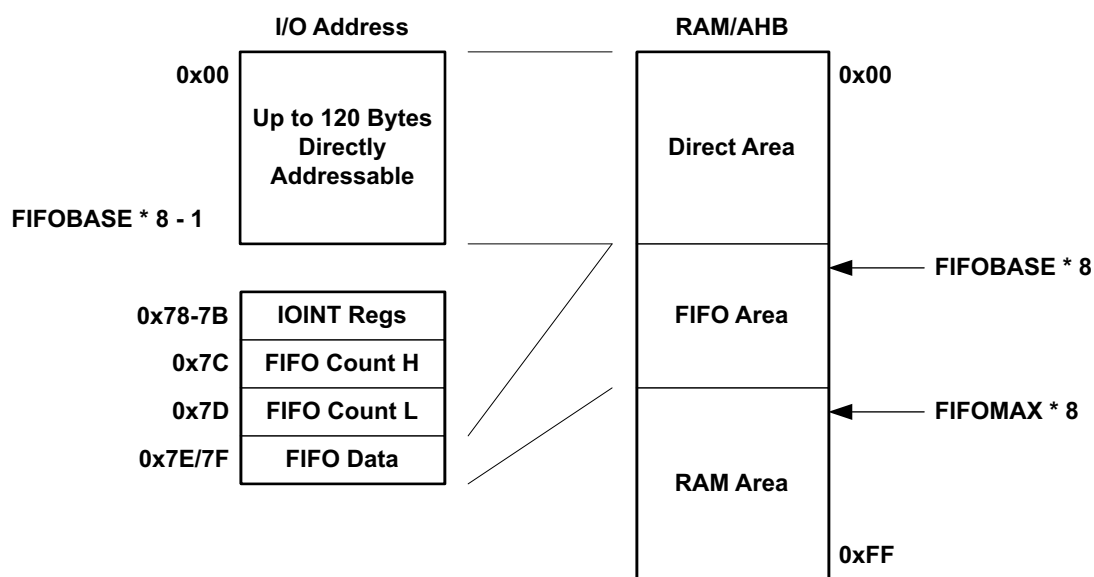


Figure 24. I²C/SPI Slave Module LRAM Addressing

6.3 Direct Area Functions

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

I/O writes to locations 0x0-0xF will set a corresponding interrupt flag in the ACC_ISR register. These locations are typically used for specific commands to the Apollo MCU. Note that not all flags need generate an actual interrupt, so small multi-byte commands may be transmitted in this area.

I/O writes to locations 0x10 to 0x4F will set a corresponding interrupt flag in the ACC_ISR 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.

The ACC_ISR register includes 32 individual write interrupts, so the REG_IOSLAVE_PRENC register provides a priority encode of these interrupts to speed software decoding. The highest priority is bit 31 (set on access to address 0x00), and the lowest priority is bit 0 (set on access to address 0x4F).

The third special area is a read-only area, which is from I/O addresses (REG_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.

6.4 FIFO Area Functions

The FIFO is used to provide very efficient flow of data from the Apollo MCU to the I/O Host processor with minimal CPU interactions. 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 I²C/SPI Slave on demand. Several hardware features support this operation.

Figure 25 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 REG_IOSLAVE_FIFOPTR_FIFOPTR register points to the next byte to be read on the I/O interface. REG_IOSLAVE_FIFOPTR_FIFOSIZ holds the current number of valid bytes in the FIFO on the I²C/SPI Slave, and FIFOCNT holds the total number of bytes in the FIFO. REG_IOSLAVE_FIFOCNT may be read at any time by the Host processor at offset 0x7C/D to determine if there is FIFO data available (and how much is currently in the FIFO).

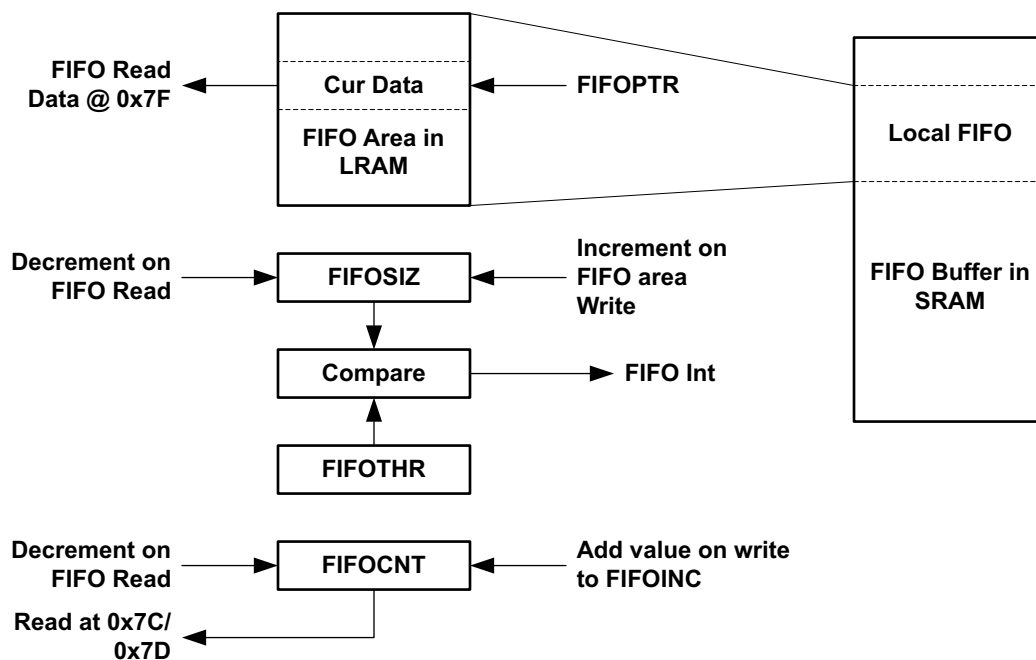


Figure 25. I²C/SPI Slave Module FIFO

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 FIFOCNT are each decremented by one. The Host can read FIFOCNT 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 I²C Slave interrupt block and in the Host interrupt block.

When FIFOSIZ drops below the configured threshold REG_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 I²C/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, 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 REG_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 I²C/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.

6.5 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 MCU supports such operation using a special control function.

If software desires to write the current sample to the top of the FIFO, it first checks the REG_IOSLAVE_FUPD_IOREAD status bit to insure that there is not a Host read operation from the FIFO underway. Once IOREAD is clear, software sets the REG_IOSLAVE_FUPD_FIFOUPD bit, writes the new sample data to the top 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.

6.6 Interface Interrupts

The CPU may also signal the Host via the IOINT interrupt, which may be connected to an Apollo MCU 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.

6.7 Host Address Space and Registers

The Host of the I/O interface can access 128 bytes in the I²C/SPI Slave in either I²C 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 I²C/SPI Slave. The R/W indicator is referring to accesses from the Host.

6.8 I²C Interface

The Apollo MCU I²C Slave interface operates as a standard slave. The device is accessed at an address configured in the REG_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 10BIT. The I²C 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 MCU I²C 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 26) and are described in the following sections.

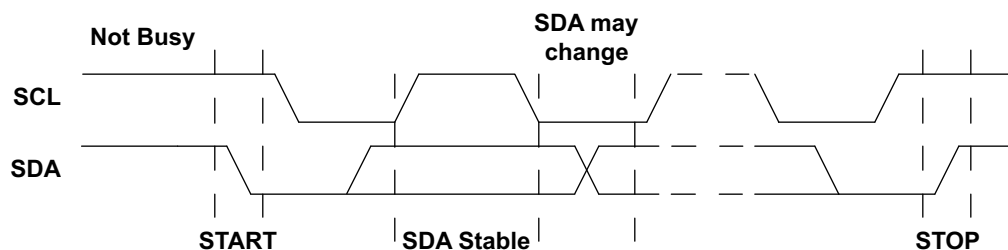


Figure 26. Basic I²C Conditions

6.8.1 Bus Not Busy

Both SDA and SCL remain high.

6.8.2 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.

6.8.3 Stop Data Transfer

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

6.8.4 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.

6.8.5 Acknowledge

Each byte of eight bits is followed by one Acknowledge (ACK) bit as shown in Figure 27. 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.

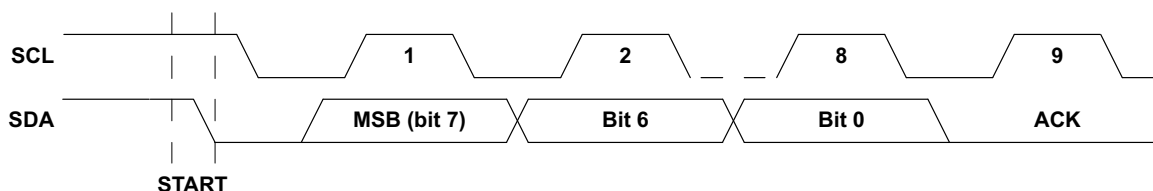


Figure 27. I²C Acknowledge

6.8.6 Address Operation

In I²C mode, the I²C/SPI Slave supports either 7-bit or 10-bit addressing, selected by the 10BIT bit in the IOSCFG Register. Figure 28 shows the operation in 7-bit mode in which the master addresses the Apollo MCU with a 7-bit address configured as 0xD2 in the I2CADDR field. After the START condition, the 7-bit address is transmitted MSB first. If this address matches the lower 7 bits of the I2CADDR field, the Apollo MCU is selected, the eighth bit indicate a write (RW = 0) or a read (RW = 1) operation and the Apollo MCU supplies the ACK. The Apollo MCU ignores all other address values and does not respond with an ACK.

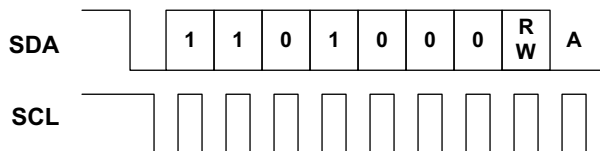


Figure 28. I²C 7-bit Address Operation

Figure 28 shows the operation with which the master addresses the Apollo 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 I²C/SPI Slave supplies the ACK. The next transfer includes the lower 8 bits of the address, and if these bits also match I2CADDR the Apollo MCU again supplies the ACK. The I²C/SPI Slave ignores all other address values and does not respond with an ACK.

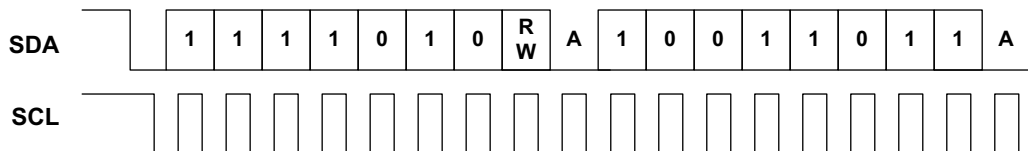


Figure 29. I²C 10-bit Address Operation

6.8.7 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 30. This value is loaded into the Address Pointer of the I²C/SPI Slave.

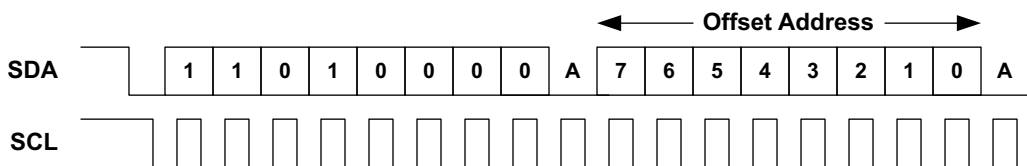


Figure 30. I²C Offset Address Transmission

6.8.8 Write Operation

In a write operation the master transmitter transmits to the Apollo MCU slave receiver. The Address Operation has a RW value of 0, and the second byte contains the Offset Address as in Figure 30. 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 31. Note that if the Address Pointer is at 0x7F, it will not increment on the write.

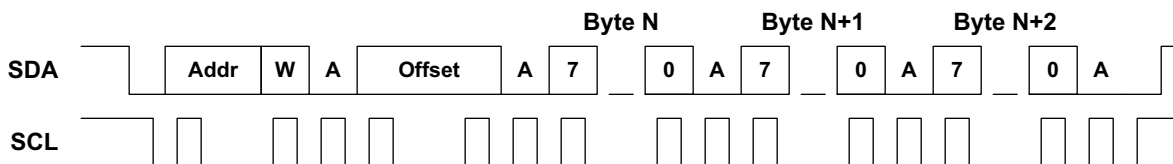


Figure 31. I²C Write Operation

6.8.9 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 Apollo MCU but with the RW bit as a 1 indicating a read operation. Figure 32 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.

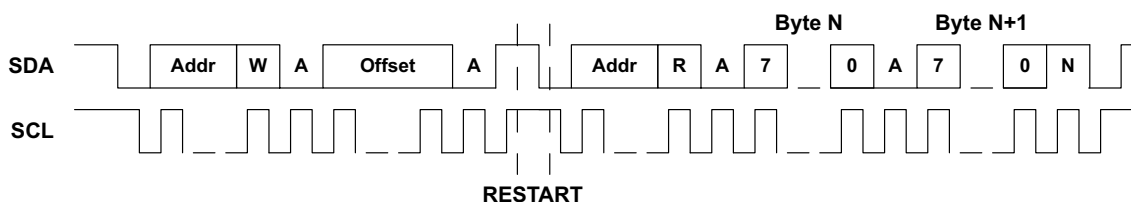


Figure 32. I²C Read Operation

6.8.10 General Address Detection

The I²C/SPI Slave may be configured to detect an I²C General Address (0x00) write. If this address is detected, the first data byte written is stored in the REG_IOSLAVE_GADATA Register and the GENAD interrupt flag is set. This allows software to create the appropriate response, which is typically to reset the I²C/SPI Slave.

6.9 SPI Interface

The I²C/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 I²C/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 I²C/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 I²C/SPI Slave supports clock frequencies up to 10 MHz, and responds to 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.

6.9.1 Write Operation

Figure 33 shows a SPI write operation. The operation is initiated when the nCE signal to the Apollo 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 I²C/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 I²C/SPI Slave. Note also that if the Address Pointer is 0x7F, it does not increment on the read.

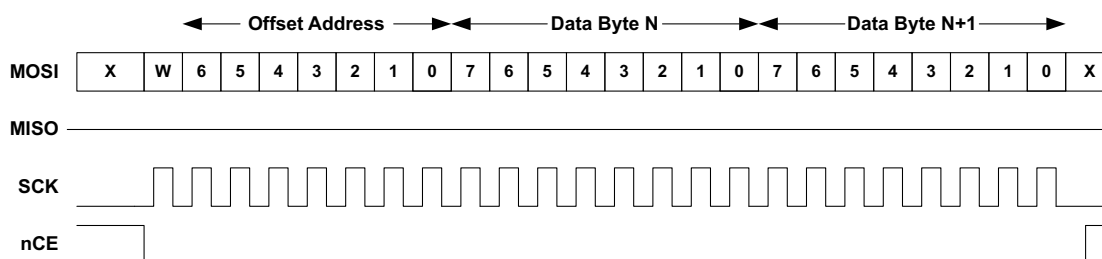


Figure 33. SPI Write Operation

6.9.2 Read Operation

Figure 34 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 I²C/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.

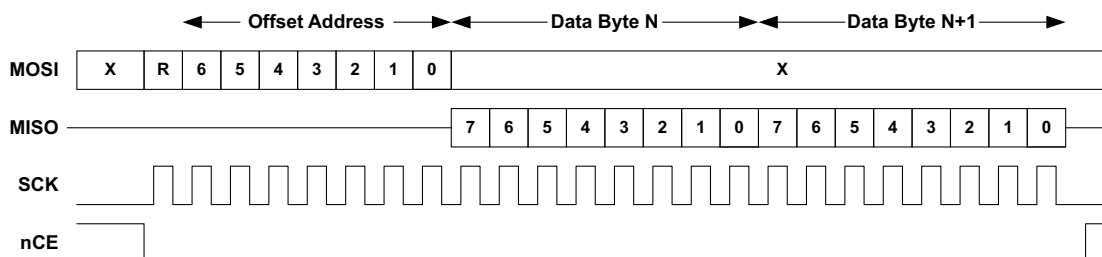


Figure 34. SPI Read Operation

6.9.3 Configuring 3-wire vs. 4-wire SPI Mode

The I²C/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 I²C/SPI Slave itself.

6.9.4 SPI Polarity and Phase

The I²C/SPI Slave supports all combinations of CPOL (clock polarity) and CPHA (data phase) in SPI mode. Figure 35 shows how these two bits affect the interface signal behavior.

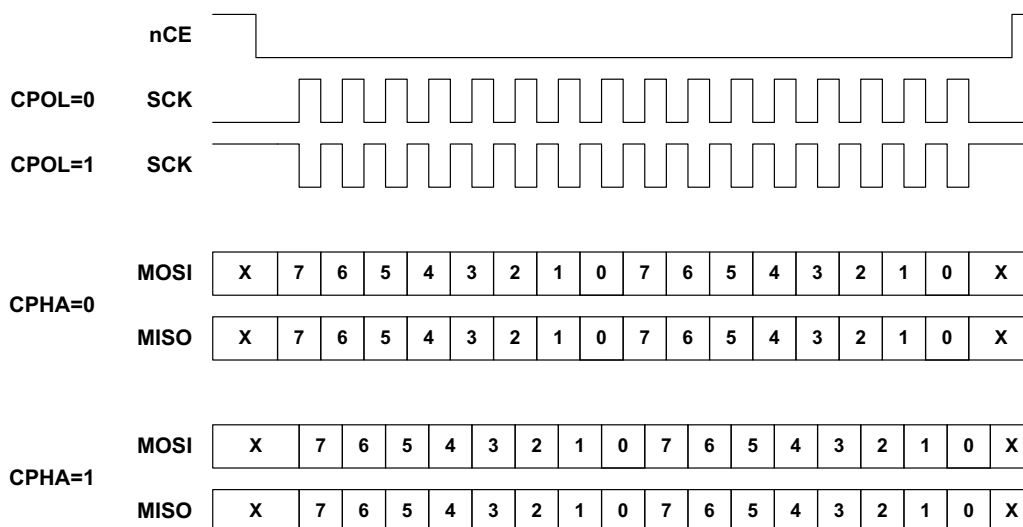


Figure 35. SPI CPOL and CPHA

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 I²C/SPI Slave has only a single SPOL bit to control the polarity. If CPOL = CPHA, REG_IOSLAVE_IOSCFG_SPOL must be set to 0. If CPOL ≠ CPHA, SPOL must be set to 1.

6.10 Bit Orientation

In both I²C and SPI modes, the I²C/SPI Slave supports data transmission either LSB first or MSB first as configured by the REG_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.

6.11 Wakeup Using the I²C/SPI Slave

The I²C/SPI Slave can continue to operate even if the Apollo MCU CPU is in Sleep or Deep Sleep mode. The hardware will enable and disable the I²C/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 I²C/SPI Slave. In this case the HFRC Oscillator must be powered up before any 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 I²C mode.

For I²C 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 Apollo 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 I²C/SPI Slave clock will start immediately when nCE goes low.

6.12 IOSLAVE Registers

I2C/SPI Slave

INSTANCE 0 BASE ADDRESS:0x50000000

6.12.1 Register Memory Map

Table 164: 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

6.12.2 IOSLAVE Registers

6.12.2.1 FIFOPTR Register

Current FIFO Pointer

OFFSET: 0x00000100

INSTANCE 0 ADDRESS: 0x50000100

Current FIFO Pointer

Table 165: FIFOPTR Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	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
RSVD																FIFOSIZ						FIFOPTR									

Table 166: 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.

6.12.2.2 FIFOCFG Register

FIFO Configuration

OFFSET: 0x00000104

INSTANCE 0 ADDRESS: 0x50000104

FIFO Configuration

Table 167: FIFOCFG Register

3	3	2	2	2	2	2	2	2	2	2	2	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
RSVD	ROBASE							RSVD							RSVD	FIFOMAX							RSVD	FIFOBASE							

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 (FIFOBASE*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).

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	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						
RSVD																						FIFOTHR															

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

6.12.2.4 FUPD Register

FIFO Update Status

OFFSET: 0x0000010C

INSTANCE 0 ADDRESS: 0x5000010C

FIFO Update Status

Table 171: FUPD Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																													IOREAD	FIFOUPD	

Table 172: 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.

6.12.2.5 FIFOCTR Register

Overall FIFO Counter

OFFSET: 0x00000110

INSTANCE 0 ADDRESS: 0x50000110

Overall FIFO Counter

Table 173: FIFOCTR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0					
RSVD																					FIFOCTR															

Table 174: FIFOCTR Register Bits

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

Table 174: FIFOCTR Register Bits

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

6.12.2.6 FIFOINC Register

Overall FIFO Counter Increment

OFFSET: 0x00000114

INSTANCE 0 ADDRESS: 0x50000114

Overall FIFO Counter Increment

Table 175: FIFOINC 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	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	
RSVD																					FIFOINC											

Table 176: 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

6.12.2.7 CFG Register

I/O Slave Configuration

OFFSET: 0x00000118

INSTANCE 0 ADDRESS: 0x50000118

I/O Slave Configuration

Table 177: CFG Register

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	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
IFCEN	RSVD											I2CADDR										RSVD				STARTRD	RSVD	LSB	SPOL	IFCSEL	

Table 178: CFG Register Bits

Bit	Name	Reset	RW	Description
31	IFCEN	0x0	RW	IOSLAVE interface enable. DIS = 0x0 - Disable the IOSLAVE EN = 0x1 - Enable the IOSLAVE
30:20	RSVD	0x0	RO	RESERVED
19:8	I2CADDR	0x0	RW	7-bit or 10-bit I2C device address.
7:5	RSVD	0x0	RO	RESERVED
4	STARTRD	0x0	RW	This bit holds the cycle to initiate an I/O RAM read. LATE = 0x0 - Initiate I/O RAM read late in each transferred byte. 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. MSB_FIRST = 0x0 - Data is assumed to be sent and received with MSB first. LSB_FIRST = 0x1 - Data is assumed to be sent and received with LSB first.
1	SPOL	0x0	RW	This bit selects SPI polarity. SPI_MODES_0_3 = 0x0 - Polarity 0, handles SPI modes 0 and 3. SPI_MODES_1_2 = 0x1 - Polarity 1, handles SPI modes 1 and 2.
0	IFCSEL	0x0	RW	This bit selects the I/O interface. I2C = 0x0 - Selects I2C interface for the IO Slave. SPI = 0x1 - Selects SPI interface for the IO Slave.

6.12.2.8 PRENC Register

I/O Slave Interrupt Priority Encode

OFFSET: 0x0000011C

INSTANCE 0 ADDRESS: 0x5000011C

I/O Slave Interrupt Priority Encode

Table 179: PRENC Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	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
RSVD																						PRENC					

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.

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
JOINTSET								RSVD						JOINTCLR	JOINT						JOINTEN										

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 set/read the IOINT interrupt enables.

6.12.2.10 GENADD Register

General Address Data

OFFSET: 0x00000124

INSTANCE 0 ADDRESS: 0x50000124

General Address Data

Table 183: GENADD Register

3	3	2	2	2	2	2	2	2	2	2	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								
RSVD																						GADATA															

Table 184: 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.

6.12.2.11 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 185: INTEN Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																										JOINTW	GENAD	FRDERR	FUNDPL	FOVFL	FSSIZE

Table 186: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5	IOINTW	0x0	RW	I2C Interrupt Write interrupt.

Table 186: 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.

6.12.2.12 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 187: INTSTAT Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																										IOINTW	GENAD	FRDERR	FUNDPL	FOVFL	FSIZE

Table 188: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5	IOINTW	0x0	RW	I2C 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.

6.12.2.14 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 191: INTSET Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																										IOINTW	GENAD	FRDERR	FUNDPL	FOVFL	FSIZE

Table 192: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED
5	IOINTW	0x0	RW	I2C 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.

6.12.2.15 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 193: REGACCINTEN Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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
REGACC																															

Table 194: REGACCINTEN Register Bits

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

6.12.2.16 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 195: REGACCINTSTAT Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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
REGACC																															

Table 196: REGACCINTSTAT Register Bits

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

6.13 Host Side Address Space and Register

6.13.1 Host Address Space and Registers

The Host of the I/O interface can access 128 bytes in the I²C/SPI Slave in either I²C 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 I²C/SPI Slave. The R/W indicator refers to accesses from the Host.

6.13.1.1 HOST_IER Register

Host Interrupt Enable

OFFSET: 0x78

This register enables the FIFO read interrupts.

Table 201: HOST_IER Register

3	3	2	2	2	2	2	2	2	2	2	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		
RSVD																								FUNDFLEN	RDERREN	SWINT5EN	SWINT4EN	SWINT3EN	SWINT2EN	SWINT1EN	SWINT0EN

Table 202: 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

6.13.1.2 *HOST_ISR Register*

Host Interrupt Status Register

OFFSET: 0x79

The host uses this register to read interrupt status.

Table 203: HOST_ISR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																								FUNDFLSTAT	RDERRSTAT	SWINT5STAT	SWINT4STAT	SWINT3STAT	SWINT2STAT	SWINT1STAT	SWINT0STAT

Table 204: 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 I0INTCTL 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 I0INTCTL 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 I0INTCTL Register.
4	SWINT4STAT	0x0	RO	This bit is set by writing a 1 to bit 28 of the I0INTCTL Register.
3	SWINT3STAT	0x0	RO	This bit is set by writing a 1 to bit 27 of the I0INTCTL Register.
2	SWINT2STAT	0x0	RO	This bit is set by writing a 1 to bit 26 of the I0INTCTL Register.
1	SWINT1STAT	0x0	RO	This bit is set by writing a 1 to bit 25 of the I0INTCTL Register.
0	SWINT0STAT	0x0	RO	This bit is set by writing a 1 to bit 24 of the I0INTCTL Register.

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

6.13.1.3 *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 205: HOST_WCR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																								FUNDFLWC	RDERRWC	SWINT5WC	SWINT4WC	SWINT3WC	SWINT2WC	SWINT1WC	SWINT0WC

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

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																								FUNDLWS	RDERRWS	SWINT5WS	SWINT4WS	SWINT3WS	SWINT2WS	SWINT1WS	SWINT0WS

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

6.13.1.5 FIFOCTRLO Register

FIFOCTR Low Byte

OFFSET: 0x7C

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

Table 209: FIFOCTRLO Register

3	3	2	2	2	2	2	2	2	2	2	2	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
RSVD																						FIFOCTRLO							

Table 210: FIFOCTRLO Register Bits

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

6.13.1.6 FIFOCTRUP Register

FIFOCTR Upper Byte

OFFSET: 0x7D

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

Table 211: FIFOCTRUP Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																														FIFOCTRUP	

Table 212: FIFOCTRUP Register Bits

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

6.13.1.7 FIFO Register

FIFO Read Data

OFFSET: 0x7F

Read this register for FIFO data.

Table 213: FIFO Register

3	3	2	2	2	2	2	2	2	2	2	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
RSVD																					FIFO										

Table 214: FIFO Register Bits

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

7. GPIO and Pad Configuration Module

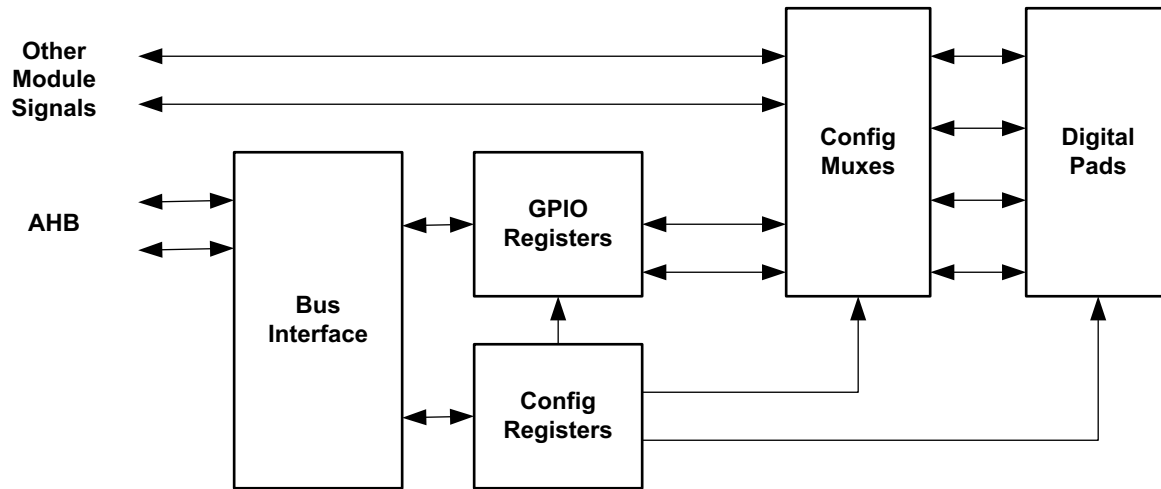


Figure 36. Block diagram for the General Purpose I/O (GPIO) Module

7.1 Functional Overview

The General Purpose I/O and Pad Configuration (GPIO) Module, shown in Figure 36, 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.

7.2 Pad Configuration Functions

The REG_GPIO_PADREGy (y = A to M) registers are used to control the function of each pad. Note that the REG_GPIO_PADKEY Register must be set to the value 0x73 in order to write the PADREGn Registers. The REG_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 215. Functions are grouped by module, with the color coding shown in Table 216. This table also defines the pad type for each configuration. The Special pad types are defined in Table 217. 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 215

The REG_GPIO_PADREGy_PADnSTRNG bit controls the drive strength of the pad. For all pads except for pad 20, REG_GPIO_PADREGy_PADnPULL bit enables a weak pull-up on the pad when set to one. For pad 20, the REG_GPIO_PADREGy_PAD20PULL bit enables a weak pull-down on the pad when set to one. The REG_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 VDDP. Pads with a (**) (pad 11) have selectable low side power switch transistors to provide ~1 Ω switches to VSS. The high side power switches are enabled by setting the REG_GPIO_PADREGy_PAD3PWRUP or REG_GPIO_PADREGy_PAD4PWRUP bits, and the 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 I²C mode, to eliminate the need for external resistors. If the pull-up is enabled by the PADnPULL bit, the REG_GPIO_PADREGy_PADnRSEL field selects the size of the pull-up resistor as shown in Table 218.

Table 215: 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 216: 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 VDDP.
**	Low-side power switch	Pads with a (**) (pad 11) have selectable low side power switch transistors to provide ~1 Ω switches to VSS.

Table 217: 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 218: I²C Pullup Resistor Selection

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

7.3 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.

7.3.1 Configuring the GPIO Functions

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

7.3.2 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 GPIORDA Register, and the current values of pads 32 to 49 are read in the GPIORDB Register. If the GPIOOnINCFG bit is set for a GPIO, it will always read as zero.

7.3.3 Writing to a GPIO Pad

The GPIO pad outputs are controlled by the GPIOWTA/B Registers and the GPIOENA/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 GPIOWTSB Register is written, the corresponding bit in GPIOWTA/B will be set if the write data is 1, otherwise the GPIOWTA/B bit will not be changed. If the GPIOWTCB Register is written, the corresponding bit in GPIOWTA/B will be cleared if the write data is 1, otherwise the GPIOWTA/B bit will not be changed.

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

7.3.4 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 selected by the GPIOOnINTP bit. 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 GPIOA_IER, GPIOA_ISR, GPIOA_WCR and GPIOB_WSR for

GPIO pads 0 to 31, and Registers GPIOB_IER, GPIOB_ISR, GPIOB_WCR and GPIOB_WSR for GPIO pads 32 to 49.

7.4 Pad Connection Summary

Figure 37 shows the detailed implementation of each pad. Each element will be described in detail.

7.4.1 Output Selection

There is multiplexer (4 or 8 input depending upon the pad) which selects the module signal to be driven to the output based on PADnFNCSEL. This implements the multiplexing shown in Table 215 for output pads. For all pads, a PADnFNCSEL value of 0x3 selects the value in the corresponding GPIOWT register bit.

7.4.2 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 216 and Table 217.

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.

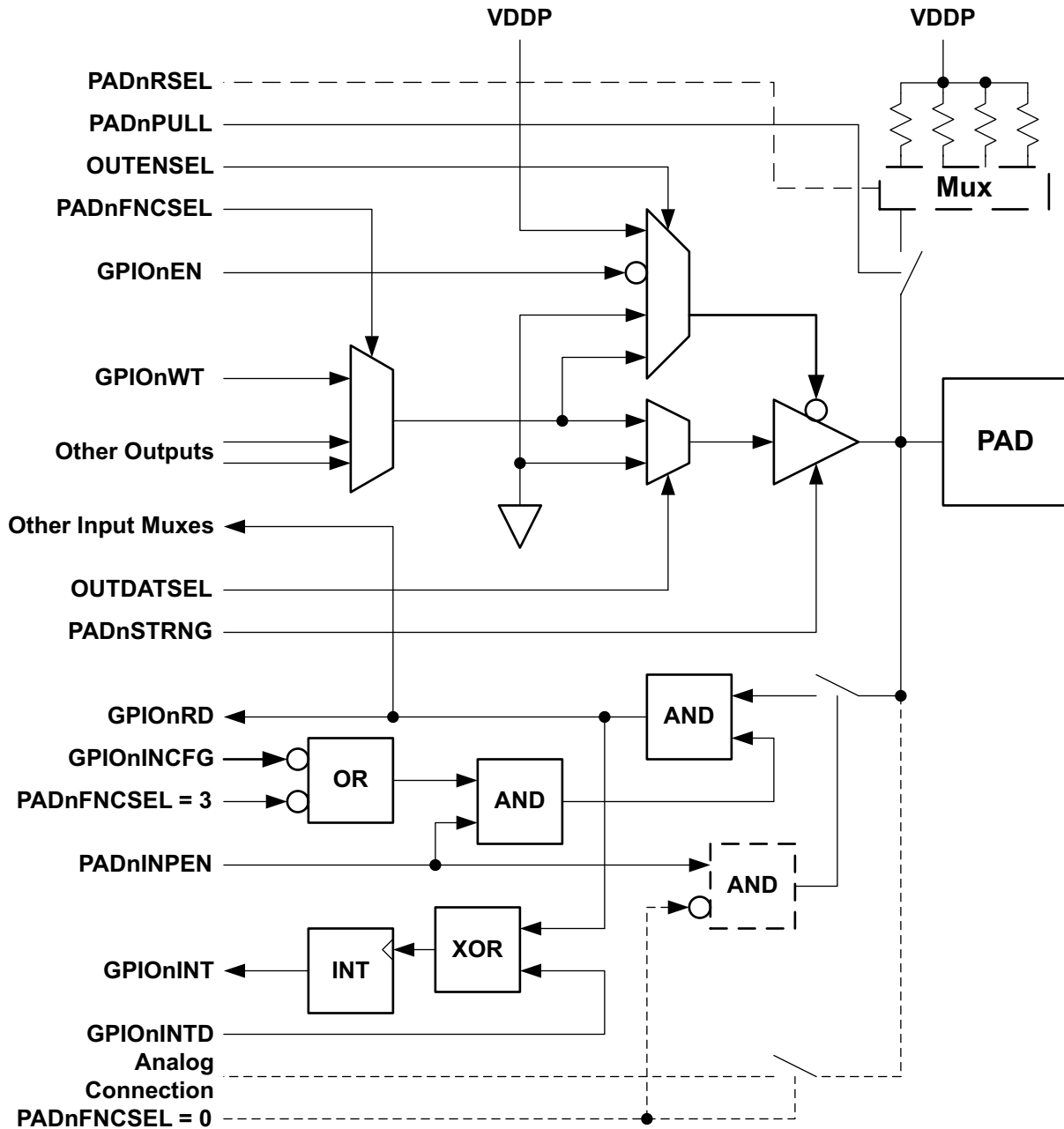


Figure 37. Pad Connection Details

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 GPIOEN bit. 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 by the PADnSTRNG bit.

7.4.3 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 215. In addition, the pad input can always be read from the GPIORD register unless the pad is configured as a GPIO (PADnFNCSEL = 0x3) and GPIOnINCFG is high, which will force the GPIORD 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 corresponding 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 I²C/SPI Slave nCE signal is driven low by the Interface Host.

7.4.4 Pull-up Control

If PADnPULL is high, a pullup resistor is connected between the pad and VDDP, except for pad 20, where PADnPULL connects the resistor to VSS rather than VDDP.

The four pads which can be I²C/SPI Master output drivers (pads 5, 6, 8 and 9) contain the additional circuitry shown with the dashed lines. In this case four different pullup resistors are selected by the PADnRSEL field.

7.4.5 Analog Pad Configuration

Pads which may have Analog connections (pads 11-19 and 29-34) include the circuitry shown with the dotted lines. 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.

7.5 Module-specific Pad Configuration

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

7.5.1 Implementing IO Master Connections

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

7.5.1.1 IO Master 0 I²C Connection

I²C 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 219. The PAD5INPEN and PAD6INPEN bits must be set. If the internal I²C 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 218. If external pullup resistors are used, PAD5PULL and PAD6PULL should be cleared.

Table 219: IO Master 0 I²C Configuration

Field	Value
PAD5FNCSEL	0
PAD6FNCSEL	0

7.5.1.2 IO Master 1 I²C Connection

I²C 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 220. The PAD8INPEN and PAD9INPEN bits must be set. If the internal I²C 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 218. If external pullup resistors are used, PAD8PULL and PAD9PULL should be cleared.

Table 220: IO Master 1 I²C Configuration

Field	Value
PAD8FNCSEL	0
PAD9FNCSEL	0

7.5.1.3 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 221. The PAD5INPEN and PAD6INPEN bits must be set. PAD5PULL, PAD6PULL and PAD7PULL should be cleared.

Table 221: IO Master 0 4-wire SPI Configuration

Field	Value
PAD5FNCSEL	1
PAD6FNCSEL	1
PAD7FNCSEL	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 222. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

Table 222: 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

7.5.1.4 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 223. The PAD8INPEN and PAD9INPEN bits must be set. PAD8PULL, PAD9PULL and PAD10PULL should be cleared.

Table 223: 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 224. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

Table 224: 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

7.5.1.5 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 225. The PAD5INPEN and PAD7INPEN bits must be set. PAD5PULL and PAD7PULL should be cleared. Pad 6 may be used for other functions.

Table 225: 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 222. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

7.5.1.6 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 226. The PAD8INPEN and PAD10INPEN bits must be set. PAD8PULL and PAD10PULL should be cleared. Pad 9 may be used for other functions.

Table 226: 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 224. The PADnINPEN and PADnPULL bits of any pad used for nCE should be cleared.

7.5.2 Implementing IO Slave Connections

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

7.5.2.1 IO Slave I²C Connection

I²C 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 227. The PAD0INPEN and PAD1INPEN bits must be set. PAD0PULL and PAD1PULL should be cleared.

Table 227: IO Slave I²C Configuration

Field	Value
PAD0FNCSEL	0
PAD1FNCSEL	0

7.5.2.2 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 228. The PAD0INPEN, PAD2INPEN and PAD3INPEN bits must be set. PAD0PULL, PAD1PULL, PAD2PULL and PAD3PULL should be cleared.

Table 228: IO Slave 4-wire SPI Configuration

Field	Value
PAD0FNCSEL	1
PAD1FNCSEL	1
PAD2FNCSEL	1
PAD3FNCSEL	1

7.5.2.3 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 229. 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 229: IO Slave 3-wire SPI Configuration

Field	Value
PAD0FNCSEL	1
PAD2FNCSEL	0
PAD3FNCSEL	1

7.5.2.4 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.

7.5.2.5 Implementing Internal I²C and SPI Loopback

The Pad Configuration module includes the capability of implementing internal loopback between the IO Master and IO Slave modules in both I²C and SPI modes. This is useful for testing the Apollo MCU before connecting to external devices. There are six loopback configurations.

7.5.2.6 IO Master 0 I²C Loopback

In this mode the IO Master 0 module drives the IO Slave in I²C mode. No pullup resistors are enabled, as a logic function implements the wired-OR I²C function, so that all relevant PADnPULL bits should be set to 0. This mode is enabled with the settings in Table 230. PADnINPEN must be set for all of the relevant pads.

Table 230: IO Master 0 I²C Loopback

Field	Value
PAD0FNCSEL	6
PAD1FNCSEL	6
PAD5FNCSEL	6
PAD6FNCSEL	6

7.5.2.7 IO Master 1 I²C Loopback

In this mode the IO Master 1 module drives the IO Slave in I²C mode. No pullup resistors are enabled, as a logic function implements the wired-OR I²C function, so that all relevant PADnPULL bits should be set to 0. This mode is enabled with the settings in Table 231. PADnINPEN must be set for all of the relevant pads.

Table 231: IO Master 1 I²C Loopback

Field	Value
PAD0FNCSEL	7
PAD1FNCSEL	7

Table 231: IO Master 1 I²C Loopback

Field	Value
PAD8FNCSEL	7
PAD9FNCSEL	7

7.5.2.8 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 232. PADnINPEN must be set for all of the relevant pads.

Table 232: IO Master 0 4-wire SPI Loopback

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

7.5.2.9 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 233. PADnINPEN must be set for all of the relevant pads.

Table 233: IO Master 1 4-wire SPI Loopback

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

7.5.2.10 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 234. PADnINPEN must be set for all of the relevant pads.

Table 234: IO Master 0 3-wire SPI Loopback

Field	Value
PAD0FNCSEL	4

Table 234: IO Master 0 3-wire SPI Loopback

Field	Value
PAD2FNCSEL	6
PAD3FNCSEL	4
PAD5FNCSEL	4
PAD7FNCSEL	6

7.5.2.11 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 235. PADnINPEN must be set for all of the relevant pads.

Table 235: IO Master 1 3-wire SPI Loopback

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

7.5.2.12 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.

7.5.3 Implementing Counter/Timer Connections

Each Counter/Timer can optionally count pulses from an input pad, or generate pulses on an output pad. Table 236 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 236: 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

7.5.4 Implementing UART Connections

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

7.5.4.1 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 237 shows the connections for TX, which should have the corresponding PADnINPEN and PADnPULL bits clear. Table 238 shows the connections for RX, which must have the corresponding PADnINPEN bit set and should have the corresponding PADnPULL bit clear.

Table 237: UART TX Configuration

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

Table 238: UART RX Configuration

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

7.5.4.2 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 239 shows the connections for RTS, which should have the corresponding PADnINPEN and PADnPULL bits clear. Table 240 shows the connections for CTS, which must have the corresponding PADnINPEN bit set and should have the corresponding PADnPULL bit clear.

Table 239: UART RTS Configuration

Field	Value	Pad
PAD5FNCSEL	2	5
PAD37FNCSEL	2	37

Table 240: UART CTS Configuration

Field	Value	Pad
PAD6FNCSEL	2	6
PAD38FNCSEL	2	38

7.5.5 Implementing GPIO Connections

Each pad of the Apollo MCU 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.

7.5.6 Implementing CLKOUT Connections

The flexible clock output of the Clock Generator module, CLKOUT, may be configured on several pads as shown in Table 241. PADnINPEN and PADnPULL should be cleared in each case.

Table 241: CLKOUT Configuration

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

7.5.7 Implementing ADC Connections

Three types of pad connections may be made for the ADC module. Up to eight pads may be configured as the analog inputs, as shown in Table 242. 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 eight selectable pad choices may be configured, as shown in Table 243. 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 242: 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 243: 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

7.5.8 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 244. The voltage to be applied to the comparator positive input are shown in Table 245. In each case PADnINPEN and PADnPULL should be cleared.

Table 244: Voltage Comparator Reference Configuration

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

Table 245: Voltage Comparator Input Configuration

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

7.5.9 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 246, and PADnINPEN and PADnPULL should be cleared for the selected pad.

Table 246: SWO Configuration

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

7.6 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.

7.6.1 Register Memory Map

Table 247: 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

Table 247: GPIO Register Map

Address(s)	Register Name	Description
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
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

7.6.2 GPIO Registers

7.6.2.1 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 248: PADREGA Register

PAD3PWRUP	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0		
RSVD																																		
PAD3FNCSEL																																		
PAD3STRNG																																		
PAD3INPEN																																		
PAD3PULL																																		
RSVD																																		
PAD2FNCSEL																																		
PAD2STRNG																																		
PAD2INPEN																																		
PAD2PULL																																		
RSVD																																		
PAD1FNCSEL																																		
PAD1STRNG																																		
PAD1INPEN																																		
PAD1PULL																																		
RSVD																																		
PAD0FNCSEL																																		
PAD0STRNG																																		
PAD0INPEN																																		
PAD0PULL																																		

Table 249: PADREGA Register Bits

Bit	Name	Reset	RW	Description
31	PAD3PWRUP	0x0	RW	Pad 3 upper power switch enable DIS = 0x0 - Power switch disabled EN = 0x1 - Power switch enabled
30	RSVD	0x0	RO	RESERVED
29:27	PAD3FNCSEL	0x3	RW	Pad 3 function select TRIG0 = 0x0 - Configure as the ADC Trigger 0 signal SLnCE = 0x1 - Configure as the IOSLAVE SPI nCE signal M1nCE4 = 0x2 - Configure as the SPI channel 4 nCE signal from IOMSTR1 GPIO3 = 0x3 - Configure as GPIO3 M0nCE = 0x4 - Configure as the IOSLAVE SPI nCE loopback signal from IOMSTR0 M1nCE = 0x5 - Configure as the IOSLAVE SPI nCE loopback signal from IOMSTR1 DIS = 0x6 - Pad disabled
26	PAD3STRNG	0x0	RW	Pad 3 drive strength. LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD3INPEN	0x0	RW	Pad 3 input enable. DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD3PULL	0x0	RW	Pad 3 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
23:22	RSVD	0x0	RO	RESERVED

Table 249: PADREGA Register Bits

Bit	Name	Reset	RW	Description
21:19	PAD2FNCSEL	0x3	RW	Pad 2 function select SLWIR3 = 0x0 - Configure as the IOSLAVE SPI 3-wire MOSI/MISO signal SLMOSI = 0x1 - Configure as the IOSLAVE SPI MOSI signal CLKOUT = 0x2 - Configure as the CLKOUT signal GPIO2 = 0x3 - Configure as GPIO2 M0MOSI = 0x4 - Configure as the IOSLAVE SPI MOSI loopback signal from IOMSTR0 M1MOSI = 0x5 - Configure as the IOSLAVE SPI MOSI loopback signal from IOMSTR1 M0WIR3 = 0x6 - Configure as the IOSLAVE SPI 3-wire MOSI/MISO loopback signal from IOMSTR0 M1WIR3 = 0x7 - Configure as the IOSLAVE SPI 3-wire MOSI/MISO loopback signal from IOMSTR1
18	PAD2STRNG	0x0	RW	Pad 2 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD2INPEN	0x0	RW	Pad 2 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD2PULL	0x0	RW	Pad 2 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:14	RSVD	0x0	RO	RESERVED
13:11	PAD1FNCSEL	0x3	RW	Pad 1 function select SLSDA = 0x0 - Configure as the IOSLAVE I2C SDA signal SLMISO = 0x1 - Configure as the IOSLAVE SPI MISO signal UARTRX = 0x2 - Configure as the UART RX signal GPIO1 = 0x3 - Configure as GPIO1 M0MISO = 0x4 - Configure as the IOSLAVE SPI MISO loopback signal from IOMSTR0 M1MISO = 0x5 - Configure as the IOSLAVE SPI MISO loopback signal from IOMSTR1 M0SDA = 0x6 - Configure as the IOSLAVE I2C SDA loopback signal from IOMSTR0 M1SDA = 0x7 - Configure as the IOSLAVE I2C SDA loopback signal from IOMSTR1
10	PAD1STRNG	0x0	RW	Pad 1 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD1INPEN	0x0	RW	Pad 1 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD1PULL	0x0	RW	Pad 1 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

Table 249: PADREGA Register Bits

Bit	Name	Reset	RW	Description
7:6	RSVD	0x0	RO	RESERVED
5:3	PAD0FNCSEL	0x3	RW	Pad 0 function select SLSCL = 0x0 - Configure as the IOSLAVE I2C SCL signal SLSCK = 0x1 - Configure as the IOSLAVE SPI SCK signal UARTTX = 0x2 - Configure as the UART TX signal GPIO0 = 0x3 - Configure as GPIO0 M0SCK = 0x4 - Configure as the IOSLAVE SPI SCK loopback signal from IOMSTR0 M1SCK = 0x5 - Configure as the IOSLAVE SPI SCK loopback signal from IOMSTR1 M0SCL = 0x6 - Configure as the IOSLAVE I2C SCL loopback signal from IOMSTR0 M1SCL = 0x7 - Configure as the IOSLAVE I2C SCL loopback signal from IOMSTR1
2	PAD0STRNG	0x0	RW	Pad 0 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD0INPEN	0x0	RW	Pad 0 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD0PULL	0x0	RW	Pad 0 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

7.6.2.2 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 250: PADREGB Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD		PAD7FNCSEL			PAD7STRNG	PAD7INPEN	PAD7PULL	PAD6RSEL		PAD6FNCSEL			PAD6STRNG	PAD6INPEN	PAD6PULL	PAD5RSEL		PAD5FNCSEL			PAD5STRNG	PAD5INPEN	PAD5PULL	PAD4PWRUP	RSVD	PAD4FNCSEL			PAD4STRNG	PAD4INPEN	PAD4PULL

Table 251: PADREGB Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD7FNCSEL	0x3	RW	Pad 7 function select M0WIR3 = 0x0 - Configure as the IOMSTR0 SPI 3-wire MOSI/MISO signal M0MOSI = 0x1 - Configure as the IOMSTR0 SPI MOSI signal CLKOUT = 0x2 - Configure as the CLKOUT signal GPIO7 = 0x3 - Configure as GPIO7 SLWIR3 = 0x6 - Configure as the IOMSTR0 SPI 3-wire MOSI/MISO loop-back signal from IOSLAVE DIS = 0x7 - Pad disabled
26	PAD7STRNG	0x0	RW	Pad 7 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD7INPEN	0x0	RW	Pad 7 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD7PULL	0x0	RW	Pad 7 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
23:22	PAD6RSEL	0x0	RW	Pad 6 pullup resistor selection. PULL1_5K = 0x0 - Pullup is ~1.5 KOhms PULL6K = 0x1 - Pullup is ~6 KOhms PULL12K = 0x2 - Pullup is ~12 KOhms PULL24K = 0x3 - Pullup is ~24 KOhms
21:19	PAD6FNCSEL	0x3	RW	Pad 6 function select M0SDA = 0x0 - Configure as the IOMSTR0 I2C SDA signal M0MISO = 0x1 - Configure as the IOMSTR0 SPI MISO signal UACTS = 0x2 - Configure as the UART CTS signal GPIO6 = 0x3 - Configure as GPIO6 SLMISO = 0x4 - Configure as the IOMSTR0 SPI MISO loopback signal from IOSLAVE SLSDA = 0x6 - Configure as the IOMSTR0 I2C SDA loopback signal from IOSLAVE DIS = 0x7 - Pad disabled
18	PAD6STRNG	0x0	RW	Pad 6 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD6INPEN	0x0	RW	Pad 6 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD6PULL	0x0	RW	Pad 6 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

Table 251: PADREGB Register Bits

Bit	Name	Reset	RW	Description
15:14	PAD5RSEL	0x0	RW	Pad 5 pullup resistor selection. PULL1_5K = 0x0 - Pullup is ~1.5 KOhms PULL6K = 0x1 - Pullup is ~6 KOhms PULL12K = 0x2 - Pullup is ~12 KOhms PULL24K = 0x3 - Pullup is ~24 KOhms
13:11	PAD5FNCSEL	0x3	RW	Pad 5 function select M0SCL = 0x0 - Configure as the IOMSTR0 I2C SCL signal M0SCK = 0x1 - Configure as the IOMSTR0 SPI SCK signal UARTS = 0x2 - Configure as the UART RTS signal GPIO5 = 0x3 - Configure as GPIO5 SLSCK = 0x4 - Configure as the IOMSTR0 SPI SCK loopback signal from IOSLAVE SLSCL = 0x6 - Configure as the IOMSTR0 I2C SCL loopback signal from IOSLAVE DIS = 0x7 - Pad disabled
10	PAD5STRNG	0x0	RW	Pad 5 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD5INPEN	0x0	RW	Pad 5 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD5PULL	0x0	RW	Pad 5 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7	PAD4PWRUP	0x0	RW	Pad 4 upper power switch enable DIS = 0x0 - Power switch disabled EN = 0x1 - Power switch enabled
6	RSVD	0x0	RO	RESERVED
5:3	PAD4FNCSEL	0x3	RW	Pad 4 function select TRIG1 = 0x0 - Configure as the ADC Trigger 1 signal SLINT = 0x1 - Configure as the IOSLAVE interrupt out signal M0nCE5 = 0x2 - Configure as the SPI channel 5 nCE signal from IOMSTR0 GPIO4 = 0x3 - Configure as GPIO4 SLINTGP = 0x4 - Configure as the IOSLAVE interrupt loopback signal to GPIO4 SWO = 0x5 - Configure as the serial wire debug SWO signal CLKOUT = 0x6 - Configure as the CLKOUT signal DIS = 0x7 - Pad disabled
2	PAD4STRNG	0x0	RW	Pad 4 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD4INPEN	0x0	RW	Pad 4 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled

Table 251: PADREGB Register Bits

Bit	Name	Reset	RW	Description
0	PAD4PULL	0x0	RW	Pad 4 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

7.6.2.3 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 252: PADREGC Register

3	3	2	2	2	2	2	2	2	2	2	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
RSVD	PAD11PWRDN	RSVD	PAD11FNCSEL	PAD11STRNG	PAD11INPEN	PAD11PULL	RSVD	PAD10FNCSEL	PAD10STRNG	PAD10INPEN	PAD10PULL	PAD9RSEL	PAD9FNCSEL	PAD9STRNG	PAD9INPEN	PAD9PULL	PAD8RSEL	PAD8FNCSEL	PAD8STRNG	PAD8INPEN	PAD8PULL							

Table 253: PADREGC Register Bits

Bit	Name	Reset	RW	Description
31	RSVD	0x0	RO	RESERVED
30	PAD11PWRDN	0x0	RW	Pad 11 lower power switch enable DIS = 0x0 - Power switch disabled EN = 0x1 - Power switch enabled
29	RSVD	0x0	RO	RESERVED
28:27	PAD11FNCSEL	0x3	RW	Pad 11 function select ANATST = 0x0 - Configure as the analog test output signal M0nCE0 = 0x1 - Configure as the SPI channel 0 nCE signal from IOMSTR0 CLKOUT = 0x2 - Configure as the CLKOUT signal GPIO11 = 0x3 - Configure as GPIO11
26	PAD11STRNG	0x0	RW	Pad 11 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength

Table 253: PADREGC Register Bits

Bit	Name	Reset	RW	Description
25	PAD11INPEN	0x0	RW	Pad 11 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD11PULL	0x0	RW	Pad 11 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
23:22	RSVD	0x0	RO	RESERVED
21:19	PAD10FNCSEL	0x3	RW	Pad 10 function select M1WIR3 = 0x0 - Configure as the IOMSTR1 SPI 3-wire MOSI/MISO signal M1MOSI = 0x1 - Configure as the IOMSTR1 SPI MOSI signal M0nCE6 = 0x2 - Configure as the SPI channel 6 nCE signal from IOMSTR0 GPIO10 = 0x3 - Configure as GPIO10 DIS = 0x6 - Pad disabled SLWIR3 = 0x7 - Configure as the IOMSTR1 SPI 3-wire MOSI/MISO loop-back signal from IOSLAVE
18	PAD10STRNG	0x0	RW	Pad 10 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD10INPEN	0x0	RW	Pad 10 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD10PULL	0x0	RW	Pad 10 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:14	PAD9RSEL	0x0	RW	Pad 9 pullup resistor selection PULL1_5K = 0x0 - Pullup is ~1.5 KOhms PULL6K = 0x1 - Pullup is ~6 KOhms PULL12K = 0x2 - Pullup is ~12 KOhms PULL24K = 0x3 - Pullup is ~24 KOhms
13:11	PAD9FNCSEL	0x3	RW	Pad 9 function select M1SDA = 0x0 - Configure as the IOMSTR1 I2C SDA signal M1MISO = 0x1 - Configure as the IOMSTR1 SPI MISO signal M0nCE5 = 0x2 - Configure as the SPI channel 5 nCE signal from IOMSTR0 GPIO9 = 0x3 - Configure as GPIO9 SLMISO = 0x5 - Configure as the IOMSTR1 SPI MISO loopback signal from IOSLAVE DIS = 0x6 - Pad disabled SLSDA = 0x7 - Configure as the IOMSTR1 I2C SDA loopback signal from IOSLAVE
10	PAD9STRNG	0x0	RW	Pad 9 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength

Table 253: PADREGC Register Bits

Bit	Name	Reset	RW	Description
9	PAD9INPEN	0x0	RW	Pad 9 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD9PULL	0x0	RW	Pad 9 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:6	PAD8RSEL	0x0	RW	Pad 8 pullup resistor selection. PULL1_5K = 0x0 - Pullup is ~1.5 KOhms PULL6K = 0x1 - Pullup is ~6 KOhms PULL12K = 0x2 - Pullup is ~12 KOhms PULL24K = 0x3 - Pullup is ~24 KOhms
5:3	PAD8FNCSEL	0x3	RW	Pad 8 function select M1SCL = 0x0 - Configure as the IOMSTR1 I2C SCL signal M1SCK = 0x1 - Configure as the IOMSTR1 SPI SCK signal M0nCE4 = 0x2 - Configure as the SPI channel 4 nCE signal from IOMSTR0 GPIO8 = 0x3 - Configure as GPIO8 SLSCK = 0x5 - Configure as the IOMSTR1 SPI SCK loopback signal from IOSLAVE DIS = 0x6 - Pad disabled SLSCL = 0x7 - Configure as the IOMSTR1 I2C SCL loopback signal from IOSLAVE
2	PAD8STRNG	0x0	RW	Pad 8 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD8INPEN	0x0	RW	Pad 8 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD8PULL	0x0	RW	Pad 8 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

7.6.2.4 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 254: PADREGD Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD	PAD15FNC-SEL			PAD15STRNG	PAD15INPEN	PAD15PULL	RSVD	PAD14FNC-SEL			PAD14STRNG	PAD14INPEN	PAD14PULL	RSVD	PAD13FNC-SEL			PAD13STRNG	PAD13INPEN	PAD13PULL	RSVD	PAD12FNC-SEL	PAD12STRNG	PAD12INPEN	PAD12PULL						

Table 255: PADREGD Register Bits

Bit	Name	Reset	RW	Description
31:30	RSVD	0x0	RO	RESERVED
29:27	PAD15FNCSEL	0x3	RW	Pad 15 function select ADC3 = 0x0 - Configure as the analog ADC input 3 M1nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR1 UARTRX = 0x2 - Configure as the UART RX signal GPIO15 = 0x3 - Configure as GPIO15 DIS = 0x7 - Pad disabled
26	PAD15STRNG	0x0	RW	Pad 15 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD15INPEN	0x0	RW	Pad 15 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD15PULL	0x0	RW	Pad 15 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
23:22	RSVD	0x0	RO	RESERVED
21:19	PAD14FNCSEL	0x3	RW	Pad 14 function select ADC2 = 0x0 - Configure as the analog ADC input 2 M1nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR1 UARTTX = 0x2 - Configure as the UART TX signal GPIO14 = 0x3 - Configure as GPIO14 DIS = 0x7 - Pad disabled
18	PAD14STRNG	0x0	RW	Pad 14 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD14INPEN	0x0	RW	Pad 14 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled

Table 255: PADREGD Register Bits

Bit	Name	Reset	RW	Description
16	PAD14PULL	0x0	RW	Pad 14 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:14	RSVD	0x0	RO	RESERVED
13:11	PAD13FNCSEL	0x3	RW	Pad 13 function select ADC1 = 0x0 - Configure as the analog ADC input 1 M1nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR1 TCTB0 = 0x2 - Configure as the input/output signal from CTIMER B0 GPIO13 = 0x3 - Configure as GPIO13 SWO = 0x6 - Configure as the serial wire debug SWO signal DIS = 0x7 - Pad disabled
10	PAD13STRNG	0x0	RW	Pad 13 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD13INPEN	0x0	RW	Pad 13 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD13PULL	0x0	RW	Pad 13 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:5	RSVD	0x0	RO	RESERVED
4:3	PAD12FNCSEL	0x3	RW	Pad 12 function select ADC0 = 0x0 - Configure as the analog ADC input 0 M1nCE0 = 0x1 - Configure as the SPI channel 0 nCE signal from IOMSTR1 TCTA0 = 0x2 - Configure as the input/output signal from CTIMER A0 GPIO12 = 0x3 - Configure as GPIO12
2	PAD12STRNG	0x0	RW	Pad 12 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD12INPEN	0x0	RW	Pad 12 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD12PULL	0x0	RW	Pad 12 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

7.6.2.5 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 256: PADREGE Register

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	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			
RSVD			PAD19FNC-SEL		PAD19STRNG	PAD19INPEN	PAD19PULL	RSVD				PAD18FNC-SEL		PAD18STRNG	PAD18INPEN	PAD18PULL	RSVD			PAD17FNC-SEL			PAD17STRNG	PAD17INPEN	PAD17PULL	RSVD				PAD16FNC-SEL		PAD16STRNG	PAD16INPEN	PAD16PULL

Table 257: PADREGE Register Bits

Bit	Name	Reset	RW	Description
31:29	RSVD	0x0	RO	RESERVED
28:27	PAD19FNCSEL	0x3	RW	Pad 19 function select CMPRF = 0x0 - Configure as the analog comparator reference signal M0nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR0 TCTB1 = 0x2 - Configure as the input/output signal from CTIMER B1 GPIO19 = 0x3 - Configure as GPIO19
26	PAD19STRNG	0x0	RW	Pad 19 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
25	PAD19INPEN	0x0	RW	Pad 19 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
24	PAD19PULL	0x0	RW	Pad 19 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
23:21	RSVD	0x0	RO	RESERVED
20:19	PAD18FNCSEL	0x3	RW	Pad 18 function select CMPIN1 = 0x0 - Configure as the analog comparator input 1 signal M0nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR0 TCTA1 = 0x2 - Configure as the input/output signal from CTIMER A1 GPIO18 = 0x3 - Configure as GPIO18
18	PAD18STRNG	0x0	RW	Pad 18 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength

Table 257: PADREGE Register Bits

Bit	Name	Reset	RW	Description
17	PAD18INPEN	0x0	RW	Pad 18 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD18PULL	0x0	RW	Pad 18 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:14	RSVD	0x0	RO	RESERVED
13:11	PAD17FNCSEL	0x3	RW	Pad 17 function select CMPIN0 = 0x0 - Configure as the analog comparator input 0 signal M0nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR0 TRIG3 = 0x2 - Configure as the ADC Trigger 3 signal GPIO17 = 0x3 - Configure as GPIO17 DIS = 0x7 - Pad disabled
10	PAD17STRNG	0x0	RW	Pad 17 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD17INPEN	0x0	RW	Pad 17 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD17PULL	0x0	RW	Pad 17 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:5	RSVD	0x0	RO	RESERVED
4:3	PAD16FNCSEL	0x3	RW	Pad 16 function select ADCREf = 0x0 - Configure as the analog ADC reference input signal M0nCE4 = 0x1 - Configure as the SPI channel 4 nCE signal from IOMSTR0 TRIG2 = 0x2 - Configure as the ADC Trigger 2 signal GPIO16 = 0x3 - Configure as GPIO16
2	PAD16STRNG	0x0	RW	Pad 16 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD16INPEN	0x0	RW	Pad 16 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD16PULL	0x0	RW	Pad 16 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

7.6.2.6 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 258: PADREGF Register

3	3	2	2	2	2	2	2	2	2	2	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
RSVD		PAD23FNC-SEL		PAD23STRNG	PAD23INPEN	PAD23PULL	RSVD		PAD22FNC-SEL	PAD22STRNG	PAD22INPEN	PAD22PULL	RSVD		PAD21FNC-SEL	PAD21STRNG	PAD21INPEN	PAD21PULL	RSVD		PAD20FNC-SEL	PAD20STRNG	PAD20INPEN	PAD20PULL				

Table 259: PADREGF Register Bits

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

Table 259: PADREGF Register Bits

Bit	Name	Reset	RW	Description
18	PAD22STRNG	0x0	RW	Pad 22 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD22INPEN	0x0	RW	Pad 22 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD22PULL	0x0	RW	Pad 22 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:13	RSVD	0x0	RO	RESERVED
12:11	PAD21FNCSEL	0x0	RW	Pad 21 function select SWDIO = 0x0 - Configure as the serial wire debug data signal M1nCE6 = 0x1 - Configure as the SPI channel 6 nCE signal from IOMSTR1 TCTB2 = 0x2 - Configure as the input/output signal from CTIMER B2 GPIO21 = 0x3 - Configure as GPIO21
10	PAD21STRNG	0x0	RW	Pad 21 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD21INPEN	0x1	RW	Pad 21 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD21PULL	0x0	RW	Pad 21 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:5	RSVD	0x0	RO	RESERVED
4:3	PAD20FNCSEL	0x0	RW	Pad 20 function select SWDCK = 0x0 - Configure as the serial wire debug clock signal M1nCE5 = 0x1 - Configure as the SPI channel 5 nCE signal from IOMSTR1 TCTA2 = 0x2 - Configure as the input/output signal from CTIMER A2 GPIO20 = 0x3 - Configure as GPIO20
2	PAD20STRNG	0x0	RW	Pad 20 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD20INPEN	0x1	RW	Pad 20 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD20PULL	0x0	RW	Pad 20 pulldown enable DIS = 0x0 - Pulldown disabled EN = 0x1 - Pulldown enabled

7.6.2.7 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 260: PADREGG Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD			PAD27FNC-SEL		PAD27STRNG	PAD27INPEN	PAD27PULL	RSVD			PAD26FNC-SEL		PAD26STRNG	PAD26INPEN	PAD26PULL	RSVD			PAD25FNC-SEL		PAD25STRNG	PAD25INPEN	PAD25PULL	RSVD			PAD24FNC-SEL		PAD24STRNG	PAD24INPEN	PAD24PULL

Table 261: PADREGG Register Bits

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

Table 261: PADREGG Register Bits

Bit	Name	Reset	RW	Description
18	PAD26STRNG	0x0	RW	Pad 26 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD26INPEN	0x0	RW	Pad 26 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD26PULL	0x0	RW	Pad 26 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:13	RSVD	0x0	RO	RESERVED
12:11	PAD25FNCSEL	0x3	RW	Pad 25 function select M0nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR0 TCTA0 = 0x2 - Configure as the input/output signal from CTIMER A0 GPIO25 = 0x3 - Configure as GPIO25
10	PAD25STRNG	0x0	RW	Pad 25 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD25INPEN	0x0	RW	Pad 25 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD25PULL	0x0	RW	Pad 25 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:5	RSVD	0x0	RO	RESERVED
4:3	PAD24FNCSEL	0x3	RW	Pad 24 function select DIS = 0x0 - Pad disabled M0nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR0 CLKOUT = 0x2 - Configure as the CLKOUT signal GPIO24 = 0x3 - Configure as GPIO24
2	PAD24STRNG	0x0	RW	Pad 24 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD24INPEN	0x0	RW	Pad 24 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD24PULL	0x0	RW	Pad 24 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

7.6.2.8 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 262: PADREGH Register

3	3	2	2	2	2	2	2	2	2	2	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
RSVD		PAD31FNC-SEL		PAD31STRNG	PAD31INPEN	PAD31PULL	RSVD		PAD30FNC-SEL	PAD30STRNG	PAD30INPEN	PAD30PULL	RSVD		PAD29FNC-SEL	PAD29STRNG	PAD29INPEN	PAD29PULL	RSVD		PAD28FNC-SEL	PAD28STRNG	PAD28INPEN	PAD28PULL				

Table 263: PADREGH Register Bits

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

Table 263: PADREGH Register Bits

Bit	Name	Reset	RW	Description
18	PAD30STRNG	0x0	RW	Pad 30 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD30INPEN	0x0	RW	Pad 30 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD30PULL	0x0	RW	Pad 30 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:13	RSVD	0x0	RO	RESERVED
12:11	PAD29FNCSEL	0x3	RW	Pad 29 function select ADC4 = 0x0 - Configure as the analog ADC input 4 signal M1nCE6 = 0x1 - Configure as the SPI channel 6 nCE signal from IOMSTR1 TCTA2 = 0x2 - Configure as the input/output signal from CTIMER A2 GPIO29 = 0x3 - Configure as GPIO29
10	PAD29STRNG	0x0	RW	Pad 29 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD29INPEN	0x0	RW	Pad 29 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD29PULL	0x0	RW	Pad 29 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:5	RSVD	0x0	RO	RESERVED
4:3	PAD28FNCSEL	0x3	RW	Pad 28 function select DIS = 0x0 - Pad disabled M1nCE5 = 0x1 - Configure as the SPI channel 5 nCE signal from IOMSTR1 TCTB1 = 0x2 - Configure as the input/output signal from CTIMER B1 GPIO28 = 0x3 - Configure as GPIO28
2	PAD28STRNG	0x0	RW	Pad 28 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD28INPEN	0x0	RW	Pad 28 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD28PULL	0x0	RW	Pad 28 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

7.6.2.9 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 264: PADREGI Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0														
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 265: PADREGI Register Bits

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

Table 265: PADREGI Register Bits

Bit	Name	Reset	RW	Description
18	PAD34STRNG	0x0	RW	Pad 34 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD34INPEN	0x0	RW	Pad 34 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD34PULL	0x0	RW	Pad 34 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:13	RSVD	0x0	RO	RESERVED
12:11	PAD33FNCSEL	0x3	RW	Pad 33 function select CMPRF1 = 0x0 - Configure as the analog comparator reference 1 signal M0nCE6 = 0x1 - Configure as the SPI channel 6 nCE signal from IOMSTR0 DIS = 0x2 - Pad disabled GPIO33 = 0x3 - Configure as GPIO33
10	PAD33STRNG	0x0	RW	Pad 33 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD33INPEN	0x0	RW	Pad 33 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD33PULL	0x0	RW	Pad 33 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:5	RSVD	0x0	RO	RESERVED
4:3	PAD32FNCSEL	0x3	RW	Pad 32 function select ADC7 = 0x0 - Configure as the analog ADC input 7 signal M0nCE5 = 0x1 - Configure as the SPI channel 5 nCE signal from IOMSTR0 TCTB3 = 0x2 - Configure as the input/output signal from CTIMER B3 GPIO32 = 0x3 - Configure as GPIO32
2	PAD32STRNG	0x0	RW	Pad 32 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD32INPEN	0x0	RW	Pad 32 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD32PULL	0x0	RW	Pad 32 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

7.6.2.10 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 266: PADREGJ Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0			
RSVD			PAD39FNC-SEL		PAD39STRNG	PAD39INPEN	PAD39PULL	RSVD				PAD38FNC-SEL		PAD38STRNG	PAD38INPEN	PAD38PULL	RSVD				PAD37FNC-SEL		PAD37STRNG	PAD37INPEN	PAD37PULL	RSVD				PAD36FNC-SEL		PAD36STRNG	PAD36INPEN	PAD36PULL

Table 267: PADREGJ Register Bits

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

Table 267: PADREGJ Register Bits

Bit	Name	Reset	RW	Description
18	PAD38STRNG	0x0	RW	Pad 38 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD38INPEN	0x0	RW	Pad 38 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD38PULL	0x0	RW	Pad 38 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:13	RSVD	0x0	RO	RESERVED
12:11	PAD37FNCSEL	0x3	RW	Pad 37 function select TRIG0 = 0x0 - Configure as the ADC Trigger 0 signal M1nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR1 UARTS = 0x2 - Configure as the UART RTS signal GPIO37 = 0x3 - Configure as GPIO37
10	PAD37STRNG	0x0	RW	Pad 37 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD37INPEN	0x0	RW	Pad 37 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD37PULL	0x0	RW	Pad 37 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:5	RSVD	0x0	RO	RESERVED
4:3	PAD36FNCSEL	0x3	RW	Pad 36 function select DIS = 0x0 - Pad disabled M1nCE1 = 0x1 - Configure as the SPI channel 1 nCE signal from IOMSTR1 UARTRX = 0x2 - Configure as the UART RX signal GPIO36 = 0x3 - Configure as GPIO36
2	PAD36STRNG	0x0	RW	Pad 36 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD36INPEN	0x0	RW	Pad 36 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD36PULL	0x0	RW	Pad 36 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

7.6.2.11 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 268: PADREGK Register

3	3	2	2	2	2	2	2	2	2	2	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
RSVD		PAD43FNC-SEL		PAD43STRNG	PAD43INPEN	PAD43PULL	RSVD		PAD42FNC-SEL	PAD42STRNG	PAD42INPEN	PAD42PULL	RSVD		PAD41FNC-SEL	PAD41STRNG	PAD41INPEN	PAD41PULL	RSVD		PAD40FNC-SEL	PAD40STRNG	PAD40INPEN	PAD40PULL				

Table 269: PADREGK Register Bits

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

Table 269: PADREGK Register Bits

Bit	Name	Reset	RW	Description
18	PAD42STRNG	0x0	RW	Pad 42 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD42INPEN	0x0	RW	Pad 42 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD42PULL	0x0	RW	Pad 42 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:13	RSVD	0x0	RO	RESERVED
12:11	PAD41FNCSEL	0x3	RW	Pad 41 function select TRIG4 = 0x0 - Configure as the ADC Trigger 4 signal DIS = 0x1 - Pad disabled SWO = 0x2 - Configure as the serial wire debug SWO signal GPIO41 = 0x3 - Configure as GPIO41
10	PAD41STRNG	0x0	RW	Pad 41 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD41INPEN	0x0	RW	Pad 41 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD41PULL	0x0	RW	Pad 41 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:5	RSVD	0x0	RO	RESERVED
4:3	PAD40FNCSEL	0x3	RW	Pad 40 function select TRIG3 = 0x0 - Configure as the ADC Trigger 3 signal UARTRX = 0x1 - Configure as the UART RX signal DIS = 0x2 - Pad disabled GPIO40 = 0x3 - Configure as GPIO40
2	PAD40STRNG	0x0	RW	Pad 40 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD40INPEN	0x0	RW	Pad 40 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD40PULL	0x0	RW	Pad 40 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

7.6.2.12 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 270: PADREGL Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0			
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 271: PADREGL Register Bits

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

Table 271: PADREGL Register Bits

Bit	Name	Reset	RW	Description
18	PAD46STRNG	0x0	RW	Pad 46 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
17	PAD46INPEN	0x0	RW	Pad 46 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
16	PAD46PULL	0x0	RW	Pad 46 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
15:13	RSVD	0x0	RO	RESERVED
12:11	PAD45FNCSEL	0x3	RW	Pad 45 function select DIS = 0x0 - Pad disabled M0nCE3 = 0x1 - Configure as the SPI channel 3 nCE signal from IOMSTR0 TCTB1 = 0x2 - Configure as the input/output signal from CTIMER B1 GPIO45 = 0x3 - Configure as GPIO45
10	PAD45STRNG	0x0	RW	Pad 45 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
9	PAD45INPEN	0x0	RW	Pad 45 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
8	PAD45PULL	0x0	RW	Pad 45 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled
7:5	RSVD	0x0	RO	RESERVED
4:3	PAD44FNCSEL	0x3	RW	Pad 44 function select TRIG7 = 0x0 - Configure as the ADC Trigger 7 signal M0nCE2 = 0x1 - Configure as the SPI channel 2 nCE signal from IOMSTR0 TCTA1 = 0x2 - Configure as the input/output signal from CTIMER A1 GPIO44 = 0x3 - Configure as GPIO44
2	PAD44STRNG	0x0	RW	Pad 44 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD44INPEN	0x0	RW	Pad 44 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD44PULL	0x0	RW	Pad 44 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

7.6.2.13 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 272: PADREGM Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																			PAD49FNC-SEL	PAD49STRNG	PAD49INPEN	PAD49PULL	RSVD				PAD48FNC-SEL	PAD48STRNG	PAD48INPEN	PAD48PULL	

Table 273: PADREGM Register Bits

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

Bit	Name	Reset	RW	Description
2	PAD48STRNG	0x0	RW	Pad 48 drive strength LOW = 0x0 - Low drive strength HIGH = 0x1 - High drive strength
1	PAD48INPEN	0x0	RW	Pad 48 input enable DIS = 0x0 - Pad input disabled EN = 0x1 - Pad input enabled
0	PAD48PULL	0x0	RW	Pad 48 pullup enable DIS = 0x0 - Pullup disabled EN = 0x1 - Pullup enabled

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
GPIO7INTD	GPIO7OUTCFG		GPIO7INCFG	GPIO6INTD	GPIO6OUTCFG		GPIO6INCFG	GPIO5INTD	GPIO5OUTCFG		GPIO5INCFG	GPIO4INTD	GPIO4OUTCFG		GPIO4INCFG	GPIO3INTD	GPIO3OUTCFG		GPIO3INCFG	GPIO2INTD	GPIO2OUTCFG		GPIO2INCFG	GPIO1INTD	GPIO1OUTCFG		GPIO1INCFG	GPIO0INTD	GPIO0OUTCFG		GPIO0INCFG

Bit	Name	Reset	RW	Description
31	GPIO7INTD	0x0	RW	GPIO7 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
30:29	GPIO7OUTCFG	0x0	RW	GPIO7 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state

Table 275: CFGA Register Bits

Bit	Name	Reset	RW	Description
28	GPIO7INCFG	0x0	RW	GPIO7 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
27	GPIO6INTD	0x0	RW	GPIO6 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
26:25	GPIO6OUTCFG	0x0	RW	GPIO6 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
24	GPIO6INCFG	0x0	RW	GPIO6 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
23	GPIO5INTD	0x0	RW	GPIO5 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
22:21	GPIO5OUTCFG	0x0	RW	GPIO5 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
20	GPIO5INCFG	0x0	RW	GPIO5 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
19	GPIO4INTD	0x0	RW	GPIO4 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
18:17	GPIO4OUTCFG	0x0	RW	GPIO4 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
16	GPIO4INCFG	0x0	RW	GPIO4 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
15	GPIO3INTD	0x0	RW	GPIO3 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition

Table 275: CFGA Register Bits

Bit	Name	Reset	RW	Description
14:13	GPIO3OUTCFG	0x0	RW	GPIO3 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
12	GPIO3INCFG	0x0	RW	GPIO3 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
11	GPIO2INTD	0x0	RW	GPIO2 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
10:9	GPIO2OUTCFG	0x0	RW	GPIO2 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
8	GPIO2INCFG	0x0	RW	GPIO2 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
7	GPIO1INTD	0x0	RW	GPIO1 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
6:5	GPIO1OUTCFG	0x0	RW	GPIO1 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO1INCFG	0x0	RW	GPIO1 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
3	GPIO0INTD	0x0	RW	GPIO0 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
2:1	GPIO0OUTCFG	0x0	RW	GPIO0 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO0INCFG	0x0	RW	GPIO0 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero

7.6.2.15 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 276: CFGB Register

3	3	2	2	2	2	2	2	2	2	2	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
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 277: CFGB Register Bits

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

Table 277: CFGB Register Bits

Bit	Name	Reset	RW	Description
23	GPIO13INTD	0x0	RW	GPIO13 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
22:21	GPIO13OUT-CFG	0x0	RW	GPIO13 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
20	GPIO13INCFG	0x0	RW	GPIO13 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
19	GPIO12INTD	0x0	RW	GPIO12 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
18:17	GPIO12OUT-CFG	0x0	RW	GPIO12 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
16	GPIO12INCFG	0x0	RW	GPIO12 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
15	GPIO11INTD	0x0	RW	GPIO11 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
14:13	GPIO11OUT-CFG	0x0	RW	GPIO11 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
12	GPIO11INCFG	0x0	RW	GPIO11 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
11	GPIO10INTD	0x0	RW	GPIO10 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
10:9	GPIO10OUT-CFG	0x0	RW	GPIO10 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state

Table 277: CFGB Register Bits

Bit	Name	Reset	RW	Description
8	GPIO10INCFG	0x0	RW	GPIO10 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
7	GPIO9INTD	0x0	RW	GPIO9 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
6:5	GPIO9OUTCFG	0x0	RW	GPIO9 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO9INCFG	0x0	RW	GPIO9 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
3	GPIO8INTD	0x0	RW	GPIO8 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
2:1	GPIO8OUTCFG	0x0	RW	GPIO8 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO8INCFG	0x0	RW	GPIO8 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero

7.6.2.16 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 278: CFGC Register

	3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0													
GPIO23INTD	GPIO23OUTCFG			GPIO23INCFG		GPIO22INTD		GPIO22OUTCFG		GPIO22INCFG		GPIO21INTD		GPIO21OUTCFG		GPIO21INCFG		GPIO20INTD		GPIO20INCFG		GPIO19INTD		GPIO19OUTCFG		GPIO19INCFG		GPIO18INTD		GPIO18OUTCFG		GPIO18INCFG		GPIO17INTD		GPIO17OUTCFG		GPIO17INCFG		GPIO16INTD		GPIO16OUTCFG		GPIO16INCFG	

Table 279: CFGC Register Bits

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

Table 279: CFGC Register Bits

Bit	Name	Reset	RW	Description
20	GPIO21INCFG	0x1	RW	GPIO21 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
19	GPIO20INTD	0x0	RW	GPIO20 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
18:17	GPIO20OUT-CFG	0x0	RW	GPIO20 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
16	GPIO20INCFG	0x1	RW	GPIO20 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
15	GPIO19INTD	0x0	RW	GPIO19 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
14:13	GPIO19OUT-CFG	0x0	RW	GPIO19 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
12	GPIO19INCFG	0x0	RW	GPIO19 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
11	GPIO18INTD	0x0	RW	GPIO18 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
10:9	GPIO18OUT-CFG	0x0	RW	GPIO18 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
8	GPIO18INCFG	0x0	RW	GPIO18 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
7	GPIO17INTD	0x0	RW	GPIO17 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition

Bit	Name	Reset	RW	Description
6:5	GPIO17OUT-CFG	0x0	RW	GPIO17 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO17INCFG	0x0	RW	GPIO17 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
3	GPIO16INTD	0x0	RW	GPIO16 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
2:1	GPIO16OUT-CFG	0x0	RW	GPIO16 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO16INCFG	0x0	RW	GPIO16 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero

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

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
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 281: CFGD Register Bits

Bit	Name	Reset	RW	Description
31	GPIO31INTD	0x0	RW	GPIO31 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
30:29	GPIO31OUT-CFG	0x0	RW	GPIO31 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
28	GPIO31INCFG	0x0	RW	GPIO31 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
27	GPIO30INTD	0x0	RW	GPIO30 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
26:25	GPIO30OUT-CFG	0x0	RW	GPIO30 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
24	GPIO30INCFG	0x0	RW	GPIO30 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
23	GPIO29INTD	0x0	RW	GPIO29 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
22:21	GPIO29OUT-CFG	0x0	RW	GPIO29 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
20	GPIO29INCFG	0x0	RW	GPIO29 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
19	GPIO28INTD	0x0	RW	GPIO28 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
18:17	GPIO28OUT-CFG	0x0	RW	GPIO28 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state

Table 281: CFGD Register Bits

Bit	Name	Reset	RW	Description
16	GPIO28INCFG	0x0	RW	GPIO28 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
15	GPIO27INTD	0x0	RW	GPIO27 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
14:13	GPIO27OUT-CFG	0x0	RW	GPIO27 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
12	GPIO27INCFG	0x0	RW	GPIO27 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
11	GPIO26INTD	0x0	RW	GPIO26 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
10:9	GPIO26OUT-CFG	0x0	RW	GPIO26 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
8	GPIO26INCFG	0x0	RW	GPIO26 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
7	GPIO25INTD	0x0	RW	GPIO25 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
6:5	GPIO25OUT-CFG	0x0	RW	GPIO25 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO25INCFG	0x0	RW	GPIO25 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
3	GPIO24INTD	0x0	RW	GPIO24 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition

Table 281: CFGD Register Bits

Bit	Name	Reset	RW	Description
2:1	GPIO24OUT-CFG	0x0	RW	GPIO24 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO24INCFG	0x0	RW	GPIO24 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero

7.6.2.18 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 282: CFGE Register

[illegible]

Table 283: CFGE Register Bits

Bit	Name	Reset	RW	Description
31	GPIO39INTD	0x0	RW	GPIO39 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
30:29	GPIO39OUT-CFG	0x0	RW	GPIO39 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
28	GPIO39INCFG	0x0	RW	GPIO39 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero

Table 283: CFGE Register Bits

Bit	Name	Reset	RW	Description
27	GPIO38INTD	0x0	RW	GPIO38 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
26:25	GPIO38OUT-CFG	0x0	RW	GPIO38 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
24	GPIO38INCFG	0x0	RW	GPIO38 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
23	GPIO37INTD	0x0	RW	GPIO37 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
22:21	GPIO37OUT-CFG	0x0	RW	GPIO37 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
20	GPIO37INCFG	0x0	RW	GPIO37 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
19	GPIO36INTD	0x0	RW	GPIO36 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
18:17	GPIO36OUT-CFG	0x0	RW	GPIO36 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
16	GPIO36INCFG	0x0	RW	GPIO36 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
15	GPIO35INTD	0x0	RW	GPIO35 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
14:13	GPIO35OUT-CFG	0x0	RW	GPIO35 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state

Table 283: CFGE Register Bits

Bit	Name	Reset	RW	Description
12	GPIO35INCFG	0x0	RW	GPIO35 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
11	GPIO34INTD	0x0	RW	GPIO34 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
10:9	GPIO34OUT-CFG	0x0	RW	GPIO34 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
8	GPIO34INCFG	0x0	RW	GPIO34 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
7	GPIO33INTD	0x0	RW	GPIO33 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
6:5	GPIO33OUT-CFG	0x0	RW	GPIO33 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO33INCFG	0x0	RW	GPIO33 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
3	GPIO32INTD	0x0	RW	GPIO32 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
2:1	GPIO32OUT-CFG	0x0	RW	GPIO32 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO32INCFG	0x0	RW	GPIO32 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero

7.6.2.19 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 284: CFGF Register

3	3	2	2	2	2	2	2	2	2	2	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
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 285: CFGF Register Bits

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

Table 285: CFGF Register Bits

Bit	Name	Reset	RW	Description
22:21	GPIO45OUT-CFG	0x0	RW	GPIO45 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
20	GPIO45INCFG	0x0	RW	GPIO45 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
19	GPIO44INTD	0x0	RW	GPIO44 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
18:17	GPIO44OUT-CFG	0x0	RW	GPIO44 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
16	GPIO44INCFG	0x0	RW	GPIO44 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
15	GPIO43INTD	0x0	RW	GPIO43 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
14:13	GPIO43OUT-CFG	0x0	RW	GPIO43 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
12	GPIO43INCFG	0x0	RW	GPIO43 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
11	GPIO42INTD	0x0	RW	GPIO42 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
10:9	GPIO42OUT-CFG	0x0	RW	GPIO42 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
8	GPIO42INCFG	0x0	RW	GPIO42 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero

Table 285: CFGF Register Bits

Bit	Name	Reset	RW	Description
7	GPIO41INTD	0x0	RW	GPIO41 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
6:5	GPIO41OUT-CFG	0x0	RW	GPIO41 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO41INCFG	0x0	RW	GPIO41 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
3	GPIO40INTD	0x0	RW	GPIO40 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
2:1	GPIO40OUT-CFG	0x0	RW	GPIO40 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO40INCFG	0x0	RW	GPIO40 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero

7.6.2.20 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 286: CFGG Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																								GPIO49INTD	GPIO49OUTCFG	GPIO49INCFG	GPIO48INTD	GPIO48OUTCFG	GPIO48INCFG		

Table 287: CFGG Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED
7	GPIO49INTD	0x0	RW	GPIO49 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
6:5	GPIO49OUT- CFG	0x0	RW	GPIO49 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
4	GPIO49INCFG	0x0	RW	GPIO49 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero
3	GPIO48INTD	0x0	RW	GPIO48 interrupt direction. INTLH = 0x0 - Interrupt on low to high GPIO transition INTHL = 0x1 - Interrupt on high to low GPIO transition
2:1	GPIO48OUT- CFG	0x0	RW	GPIO48 output configuration. DIS = 0x0 - Output disabled PUSHPULL = 0x1 - Output is push-pull OD = 0x2 - Output is open drain TS = 0x3 - Output is tri-state
0	GPIO48INCFG	0x0	RW	GPIO48 input enable. READ = 0x0 - Read the GPIO pin data RDZERO = 0x1 - Readback will always be zero

7.6.2.21 PADKEY Register

Key Register for all pad configuration registers

OFFSET: 0x00000060

INSTANCE 0 ADDRESS: 0x40010060

Key Register for all pad configuration registers

Table 288: PADKEY Register

3	3	2	2	2	2	2	2	2	2	2	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
PADKEY																															

Table 289: PADKEY Register Bits

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

7.6.2.22 RDA Register

GPIO Input Register A

OFFSET: 0x00000080

INSTANCE 0 ADDRESS: 0x40010080

GPIO Input Register A

Table 290: RDA Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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
RDA																															

Table 291: RDA Register Bits

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

7.6.2.23 RDB Register

GPIO Input Register B

OFFSET: 0x00000084

INSTANCE 0 ADDRESS: 0x40010084

GPIO Input Register B

Table 292: RDB Register

3	3	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	0	0	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
RSVD												RDB																			

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

3	3	2	2	2	2	2	2	2	2	2	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
WTA																															

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

3	3	2	2	2	2	2	2	2	2	2	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
RSVD												WTB																			

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

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	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
WTSA																									

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

3	3	2	2	2	2	2	2	2	2	2	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
RSVD												WTSB															

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

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	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
WTCA																									

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

3	3	2	2	2	2	2	2	2	2	2	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
RSVD												WTCB																			

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

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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
ENA																															

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

3	3	2	2	2	2	2	2	2	2	2	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
RSVD													ENB																		

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

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	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
ENSA																															

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

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	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
RSVD													ENSB																		

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

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	0	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	0
ENCA																																

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

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	0	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
RSVD													ENCB																		

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

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

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
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

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.

Table 319: INT0EN Register Bits

Bit	Name	Reset	RW	Description
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	GPIO18 interrupt.
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.
3	GPIO3	0x0	RW	GPIO3 interrupt.
2	GPIO2	0x0	RW	GPIO2 interrupt.
1	GPIO1	0x0	RW	GPIO1 interrupt.
0	GPIO0	0x0	RW	GPIO0 interrupt.

7.6.2.37 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 320: INT0STAT Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
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 321: 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.

Table 321: INT0STAT Register Bits

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

7.6.2.38 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.

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
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

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 323: INT0CLR 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.

7.6.2.39 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 324: INT0SET Register

3	3	2	2	2	2	2	2	2	2	2	2	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
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 325: 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.
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	GPIO18 interrupt.
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 325: INT0SET 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.

7.6.2.40 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 326: INT1EN Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD														GPIO49	GPIO48	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32

Table 327: INT1EN Register Bits

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17	GPIO49	0x0	RW	GPIO49 interrupt.

Table 327: INT1EN Register Bits

Bit	Name	Reset	RW	Description
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.

7.6.2.41 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.

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														GPI09	GPI08	GPI07	GPI06	GPI05	GPI04	GPI03	GPI02	GPI01	GPI00										

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 329: INT1STAT Register Bits

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

7.6.2.42 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 330: INT1CLR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD														GPIO49	GPIO48	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32

Table 331: 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.

Table 331: INT1CLR 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.

7.6.2.43 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 332: INT1SET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD														GPIO49	GPIO48	GPIO47	GPIO46	GPIO45	GPIO44	GPIO43	GPIO42	GPIO41	GPIO40	GPIO39	GPIO38	GPIO37	GPIO36	GPIO35	GPIO34	GPIO33	GPIO32

Table 333: 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.

Table 333: INT1SET Register Bits

Bit	Name	Reset	RW	Description
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.

8. Clock Generator and Real Time Clock Module

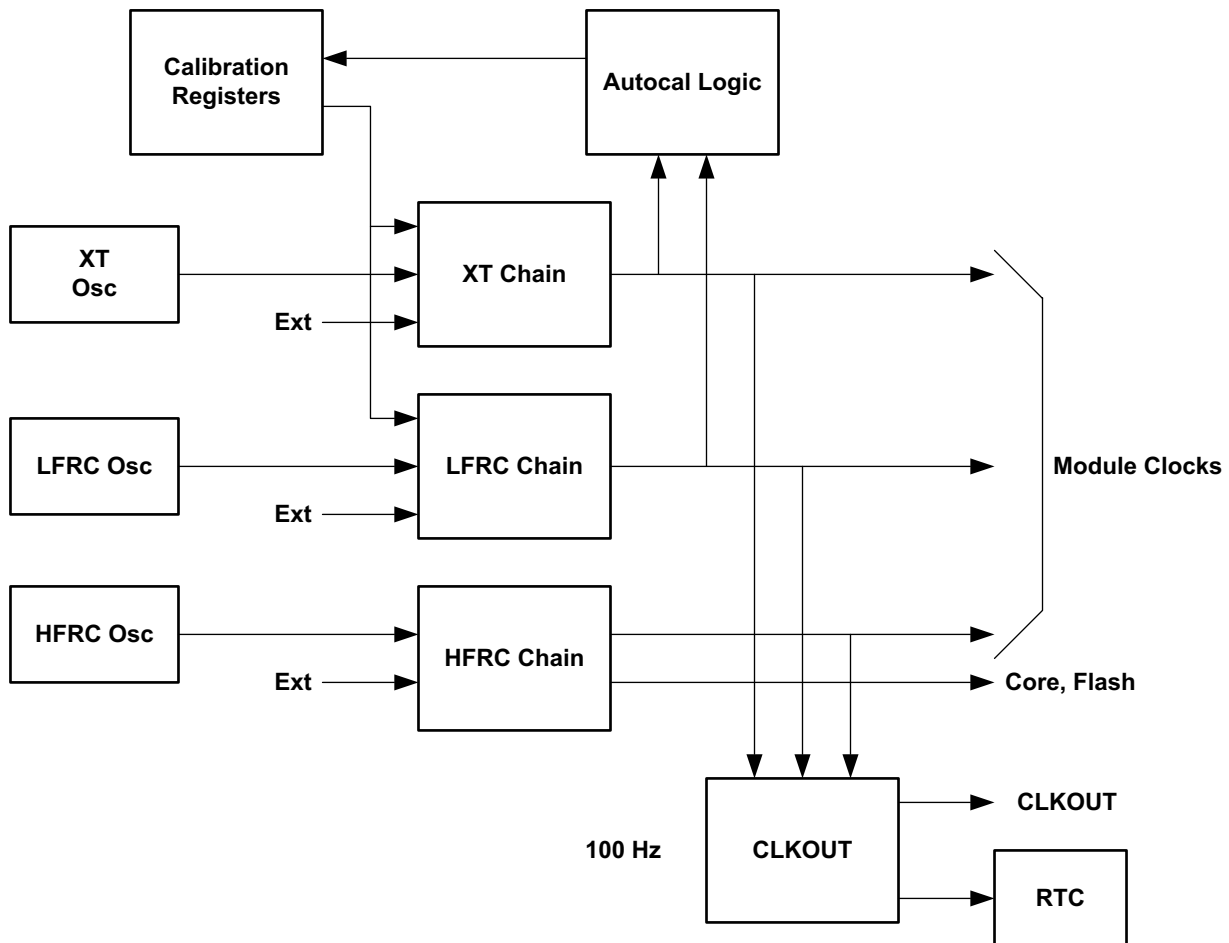


Figure 38. Block diagram for the Clock Generator and Real Time Clock Module

8.1 Clock Generator

8.1.1 Functional Overview

The Clock Generator Module, shown in Figure 38, supplies all clocks required by the Apollo MCU. 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 configurable clock CLKOUT, generated from any of the oscillators, may be configured and driven onto an external pin. Each oscillator may be replaced with an external clock input for test or measurement purposes. CLKOUT also drives the Real Time Clock (RTC) Module, which is described in Sections 8.2 to 8.2.8.

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 I²C/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.

8.1.2 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.

8.1.2.1 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 REG_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 ($\pm 25\%$).

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 MCU which use the LFRC.

Note that since the 512 Hz LFRC clock is calibrated, the 1024 Hz LFRC is an uncalibrated clock. This may be a useful selection in some cases.

8.1.2.2 LFRC Calibration Process

The LFRC oscillator calibration value is determined by the following process:

1. Set the CALRC field to 0 to insure calibration is not occurring.
2. Select the LFRC oscillator by setting the REG_CLKGEN_OCTRL_OSEL bit to 1.
3. Select the LFRC or a division of it on a CLKOUT pad.
4. Measure the frequency Fmeas at the CLKOUT pad.
5. Compute the adjustment value required in ppm as $((F_{nom} - F_{meas}) * 1000000) / F_{meas} = PAdj$
6. Compute the adjustment value in steps as $PAdj / (1000000 / 2^{19}) = PAdj / (1.90735) = Adj$
7. If $Adj < -249,955$, the LFRC frequency is too high to be calibrated
8. Else if $Adj < 249,954$, set CALRC = Adj
9. Else the LFRC frequency is too low to be calibrated

8.1.3 High Precision XT Oscillator (XT)

The high accuracy XT Oscillator is tuned to an external 32,768 Hz crystal, and has a nominal frequency of 32,768 Hz. 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.

8.1.3.1 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 REG_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 ($\pm 0.1\%$).

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 MCU which use the XT.

Note that 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.

8.1.3.2 XT Calibration Process

The XT Oscillator calibration value is determined by the following process:

1. Set the CALXT register field to 0 to insure calibration is not occurring.
2. Select the XT oscillator by setting the REG_CLKGEN_OCTRL_OSEL bit to 0.
3. Select the XT or a division of it on a CLKOUT pad.
4. Measure the frequency Fmeas at the CLKOUT pad.
5. Compute the adjustment value required in ppm as $((F_{nom} - F_{meas}) * 1000000) / F_{meas} = P_{adj}$
6. Compute the adjustment value in steps as $P_{adj} / (1000000 / 2^{19}) = P_{adj} / (0.9535) = Adj$
7. If $Adj < -976$, the XT frequency is too high to be calibrated
8. Else if $Adj < 975$, set $CALXT = Adj$
9. Else the XT frequency is too low to be calibrated.

8.1.4 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 MCU 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.

8.1.5 HFRC Autoadjustment

In some applications it is important that the HFRC frequency be more accurate than the $\pm 2\%$ variation typically seen, particularly in cases where the temperature may vary widely. A good example of this is in cases where the Apollo MCU 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 Autoadjustment is provided.

During Autoadjustment, the number of HFRC cycles which occur in one 32,768 Hz 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 REG_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.

Autoadjustment 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 REG_CLKGEN_HFTUNERB Register. Autoadjustment is enabled loading the repeat frequency value into the HFADJCK field and then setting the REG_CLKGEN_HFADJ_HFADJEN bit.

Autoadjustment cycles will occur continuously if both the XT and the HFRC are enabled. If either oscillator is disabled, Autoadjustment cycles will then occur at intervals determined by the REG_CLKGEN_HFADJ_HFADJCK field, as shown in the register description in Section 8.1.11.2.9 on page 236. 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 Autoadjustment 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 REG_CLKGEN_HFADJ_HFWARMUP field as defined in the Register document. Once the HFRC is stable, the HFRC is enabled and several Autoadjustments 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.

8.1.6 CLKOUT selection

A wide variety of internal clocks may be selected for the CLKOUT function, which is either driven onto one of a group of external pads or used in the internal frequency measurement function. The clock is selected by the REG_CLKGEN_CCTRL_CKSEL field as shown in the Register document and enabled by the REG_CLKGEN_CCTRL_CKEN bit. Entries with (NE) do not specifically enable the selected oscillator, but will drive a clock if the oscillator is enabled by another logic function. Other entries will enable the necessary oscillator if it is not already enabled by another module.

8.1.7 Frequency Measurement

The Autocalibration logic may be used to measure the frequency of an internal clock signal relative to either the XT Oscillator frequency or the frequency of an external clock. The following steps are required to perform this measurement:

1. Set the REG_CLKGEN_OCTRL_ACAL field to 000.
2. Clear the ACC interrupt flag.
3. Select the clock to be measured with the CKSEL REG_CLKGEN_CCTRL_CKSEL field.
4. Set ACAL to 110 if the XT Oscillator is to be the reference.
5. Wait for the ACC interrupt flag to be set.
6. Read the REG_CLKGEN_ACALCTR field. This will contain the number of reference clocks which occurred during one cycle of the clock selected on CLKOUT.
7. 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.

8.1.8 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 REG_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.

8.1.9 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 MCU detects an Oscillator Failure. The OF flag is set when an Oscillator Failure occurs, and is also set when the Apollo MCU 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 MCU 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 MCU 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.

8.1.10 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.

8.1.11 CLKGEN Registers

Clock Generator

INSTANCE 0 BASE ADDRESS:0x40004000

8.1.11.1 Register Memory Map

Table 334: 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

8.1.11.2 CLKGEN Registers

8.1.11.2.1 CALXT Register

XT Oscillator Control

OFFSET: 0x00000000**INSTANCE 0 ADDRESS: 0x40004000**

XT Oscillator Control

Table 335: CALXT Register

3	3	2	2	2	2	2	2	2	2	2	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						
RSVD																				CALXT																	

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10:0	CALXT	0x0	RW	XT Oscillator calibration value

3	3	2	2	2	2	2	2	2	2	2	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
RSVD													CALRC																		

Bit	Name	Reset	RW	Description
31:18	RSVD	0x0	RO	RESERVED
17:0	CALRC	0x0	RW	LFRC Oscillator calibration value

3	3	2	2	2	2	2	2	2	2	2	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				
RSVD								ACALCTR																											

Table 340: ACALCTR Register Bits

Bit	Name	Reset	RW	Description
31:24	RSVD	0x0	RO	RESERVED
23:0	ACALCTR	0x0	RO	Autocalibration Counter result.

8.1.11.2.4 OCTRL Register

Oscillator Control

OFFSET: 0x0000000C**INSTANCE 0 ADDRESS: 0x4000400C**

Oscillator Control

Table 341: OCTRL Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																					ACAL		OSEL	FOS	RSVD				STOPRC	STOPXT	

Table 342: OCTRL Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10:8	ACAL	0x0	RW	Autocalibration control DIS = 0x0 - Disable Autocalibration 1024SEC = 0x2 - Autocalibrate every 1024 seconds 512SEC = 0x3 - Autocalibrate every 512 seconds XTFREQ = 0x6 - Frequency measurement using XT EXTREQ = 0x7 - Frequency measurement using external clock
7	OSEL	0x0	RW	Selects the RTC oscillator (1 => LFRC, 0 => XT) RTC_XT = 0x0 - RTC uses the XT RTC_LFRC = 0x1 - RTC uses the LFRC
6	FOS	0x0	RW	Oscillator switch on failure function DIS = 0x0 - Disable the oscillator switch on failure function EN = 0x1 - Enable the oscillator switch on failure function
5:2	RSVD	0x0	RO	RESERVED

Bit	Name	Reset	RW	Description
1	STOPRC	0x0	RW	Stop the LFRC Oscillator to the RTC EN = 0x0 - Enable the LFRC Oscillator to drive the RTC STOP = 0x1 - Stop the LFRC Oscillator from driving the RTC
0	STOPXT	0x0	RW	Stop the XT Oscillator to the RTC EN = 0x0 - Enable the XT Oscillator to drive the RTC STOP = 0x1 - Stop the XT Oscillator from driving the RTC

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																								CKEN	RSVD	CKSEL					

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED
7	CKEN	0x0	RW	Enable the CLKOUT signal DIS = 0x0 - Disable CLKOUT EN = 0x1 - Enable CLKOUT
6	RSVD	0x0	RO	RESERVED

Table 344: 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. LFRC = 0x0 - LFRC XT_DIV2 = 0x1 - XT / 2 XT_DIV4 = 0x2 - XT / 4 XT_DIV8 = 0x3 - XT / 8 XT_DIV16 = 0x4 - XT / 16 XT_DIV32 = 0x5 - XT / 32 RTC_1Hz = 0x10 - 1 Hz as selected in RTC XT_DIV2M = 0x16 - XT / 2²¹ XT = 0x17 - XT CG_100Hz = 0x18 - 100 Hz as selected in CLKGEN HFRC = 0x19 - HFRC HFRC_DIV2 = 0x1A - HFRC / 2 HFRC_DIV4 = 0x1B - HFRC / 4 HFRC_DIV8 = 0x1C - HFRC / 8 HFRC_DIV32 = 0x1D - HFRC / 32 HFRC_DIV64 = 0x1E - HFRC / 64 HFRC_DIV128 = 0x1F - HFRC / 128 HFRC_DIV256 = 0x20 - HFRC / 256 FLASH_CLK = 0x22 - Flash Clock LFRC_DIV2 = 0x23 - LFRC / 2 LFRC_DIV32 = 0x24 - LFRC / 32 LFRC_DIV512 = 0x25 - LFRC / 512 LFRC_DIV32K = 0x26 - LFRC / 32768 XT_DIV256 = 0x27 - XT / 256 XT_DIV8K = 0x28 - XT / 8192 XT_DIV64K = 0x29 - XT / 2¹⁶ ULFRC_DIV16 = 0x2A - Uncal LFRC / 16 ULFRC_DIV128 = 0x2B - Uncal LFRC / 128 ULFRC_1Hz = 0x2C - Uncal LFRC / 1024 ULFRC_DIV4K = 0x2D - Uncal LFRC / 4096 ULFRC_DIV1M = 0x2E - Uncal LFRC / 2²⁰ HFRC_DIV64K = 0x2F - HFRC / 2¹⁶ HFRC_DIV16M = 0x30 - HFRC / 2²⁴ LFRC_DIV2M = 0x31 - LFRC / 2²⁰ HFRCNE = 0x32 - HFRC (not autoenabled) HFRCNE_DIV8 = 0x33 - HFRC / 8 (not autoenabled) XTNE = 0x35 - XT (not autoenabled) XTNE_DIV16 = 0x36 - XT / 16 (not autoenabled) LFRCNE_DIV32 = 0x37 - LFRC / 32 (not autoenabled) LFRCNE = 0x39 - LFRC (not autoenabled) - Default for undefined values

Table 348: CCTRL Register Bits

Bit	Name	Reset	RW	Description
3	MEMSEL	0x0	RW	Flash Clock divisor HFRC_DIV25 = 0x0 - Flash Clock is HFRC / 25 HFRC_DIV45 = 0x1 - Flash Clock is HFRC / 45
2:0	CORESEL	0x7	RW	Core Clock divisor HFRC = 0x0 - Core Clock is HFRC HFRC_DIV2 = 0x1 - Core Clock is HFRC / 2 HFRC_DIV3 = 0x2 - Core Clock is HFRC / 3 HFRC_DIV4 = 0x3 - Core Clock is HFRC / 4 HFRC_DIV5 = 0x4 - Core Clock is HFRC / 5 HFRC_DIV6 = 0x5 - Core Clock is HFRC / 6 HFRC_DIV7 = 0x6 - Core Clock is HFRC / 7 HFRC_DIV8 = 0x7 - Core Clock is HFRC / 8

8.1.11.2.8 STATUS Register

Clock Generator Status

OFFSET: 0x0000001C

INSTANCE 0 ADDRESS: 0x4000401C

Clock Generator Status

Table 349: STATUS Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																												OSCF	OMODE		

Table 350: 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)

8.1.11.2.9 HFADJ Register

HFRC Adjustment

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x40004020

HFRC Adjustment

Table 351: HFADJ Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD												HFWMUP	HFXTADJ										RSVD					HFADJCK		HFADJEN	

Table 352: HFADJ Register Bits

Bit	Name	Reset	RW	Description
31:20	RSVD	0x0	RO	RESERVED
19	HFARMUP	0x0	RW	XT warmup period for HFRC adjustment 1SEC = 0x0 - Autoadjust XT warmup period = 1-2 seconds 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 4SEC = 0x0 - Autoadjust repeat period = 4 seconds 16SEC = 0x1 - Autoadjust repeat period = 16 seconds 32SEC = 0x2 - Autoadjust repeat period = 32 seconds 64SEC = 0x3 - Autoadjust repeat period = 64 seconds 128SEC = 0x4 - Autoadjust repeat period = 128 seconds 256SEC = 0x5 - Autoadjust repeat period = 256 seconds 512SEC = 0x6 - Autoadjust repeat period = 512 seconds 1024SEC = 0x7 - Autoadjust repeat period = 1024 seconds
0	HFADJEN	0x0	RW	HFRC adjustment control DIS = 0x0 - Disable the HFRC adjustment EN = 0x1 - Enable the HFRC adjustment

8.1.11.2.10 HFVAL Register

HFADJ readback

OFFSET: 0x00000024

INSTANCE 0 ADDRESS: 0x40004024

HFADJ readback

Table 353: HFVAL Register

3	3	2	2	2	2	2	2	2	2	2	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				
RSVD																				HFTUNERB															

Table 354: HFVAL Register Bits

Bit	Name	Reset	RW	Description
31:11	RSVD	0x0	RO	RESERVED
10:0	HFTUNERB	0x0	RO	Current HFTUNE value

8.1.11.2.11 CLOCKEN Register

Clock Enable Status

OFFSET: 0x00000028

INSTANCE 0 ADDRESS: 0x40004028

Clock Enable Status

Table 355: CLOCKEN Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	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
CLOCKEN																															

Table 356: CLOCKEN Register Bits

Bit	Name	Reset	RW	Description
31:0	CLOCKEN	0x0	RO	Clock enable status

8.1.11.2.12 *UARTEN Register*

UART Enable

OFFSET: 0x0000002C**INSTANCE 0 ADDRESS: 0x4000402C**

UART Enable

Table 357: UARTEN Register

3	3	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	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	
RSVD																																UARTEN

Table 358: UARTEN Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED
0	UARTEN	0x0	RW	UART system clock control DIS = 0x0 - Disable the UART system clock EN = 0x1 - Enable the UART system clock

8.1.11.2.13 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 359: INTEN Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																											ALM	OF	ACC	ACF	

Table 360: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	RESERVED

Table 360: INTEN Register Bits

Bit	Name	Reset	RW	Description
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

8.1.11.2.14 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 361: INTSTAT Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																												ALM	OF	ACC	ACF

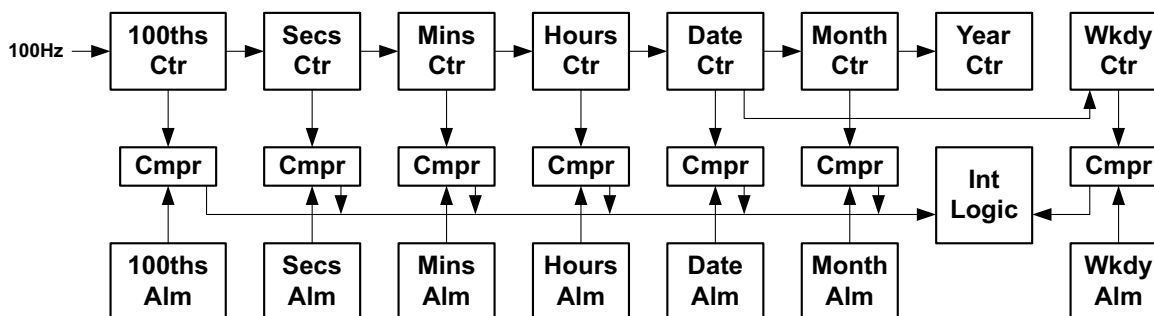
Table 362: INTSTAT Register Bits

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

Table 366: 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

8.2 Real Time Clock


Figure 39. Block diagram for the Real Time Clock Module

8.2.1 RTC Functional Overview

The Real Time Clock (RTC) Module, shown in Figure 39, 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.

8.2.2 Calendar Counters

The real time is held in a set of eight Calendar Counters, which hold the current 1/100th of a second (REG_CLK_GEN_CTRLOW_CTR100), the current second (REG_CLK_GEN_CTRLOW_CTRSEC), the minute (REG_CLK_GEN_CTRLOW_CTRMIN), the hour (REG_CLK_GEN_CTRLOW_CTRHR), the current day of the month (REG_CLK_GEN_CTRUP_CTRDATE), the current day of the week (REG_CLK_GEN_CTRUP_CTRWKDY), the current month (REG_CLK_GEN_CTRUP_CTRMO), the current year (REG_CLK_GEN_CTRUP_CTRYR) and the current century (REG_CLK_GEN_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 REG_CLK_GEN_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 REG_CLK_GEN_RTCCTL_RSTOP bit. This may be used in modes like Stopwatch to precisely start and stop the Calendar Counters.

8.2.3 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 REG_CLK_GEN_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.

8.2.4 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 (REG_CLK_GEN_ALMLOW_ALM100), second (REG_CLK_GEN_ALMLOW_ALMSEC), minute (REG_CLK_GEN_ALMLOW_ALMMIN), hour (REG_CLK_GEN_ALMLOW_ALMHR), day of the month (REG_CLK_GEN_ALMUP_ALMDATE), day of the week (REG_CLK_GEN_ALMUP_ALMWKDY) and month (REG_CLK_GEN_ALMUP_ALMMO) Calendar Counters. The comparison is controlled by the REG_CLK_GEN_RTCCTL_RPT field and the REG_CLK_GEN_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 367: Alarm RPT Function

RPT Value	ALM100	Interval	Comparison
000	0xnn	Disabled	None
000	0xnn	Every year	100 th , second, minute, hour, day, month
000	0xnn	Every month	100 th , second, minute, hour, day
001	0xnn	Every week	100 th , second, minute, hour, weekday
010	0xnn	Every day	100 th , second, minute, hour
011	0xnn	Every hour	100 th , second, minute
100	0xnn	Every minute	100 th , second
101	0xnn	Every second	100 th
110	0xFn	Every 10 th	Upper nibble of 100 th
111	0xFF	Every 100 th	None

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.

8.2.5 12/24 Hour Mode

If the REG_CLK_GEN_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.

8.2.6 Century Control and Leap Year Management

The REG_CLK_GEN_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 REG_CLK_GEN_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.

8.2.7 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.

8.2.8 RTC Registers

Real Time Clock

INSTANCE 0 BASE ADDRESS:0x40004000

8.2.8.1 Register Memory Map

Table 368: RTC Register Map

Address(s)	Register Name	Description
0x40004040	CTRL0W	RTC Counters Lower
0x40004044	CTRU0P	RTC Counters Upper
0x40004048	ALML0W	RTC Alarms Lower
0x4000404C	ALMU0P	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

8.2.8.2.4 ALMUP Register

RTC Alarms Upper

OFFSET: 0x0000004C

INSTANCE 0 ADDRESS: 0x4000404C

RTC Alarms Upper

Table 375: ALMUP Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD												ALMWKDY	RSVD		ALMMO			RSVD	ALMDATE												

Table 376: 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

8.2.8.2.6 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 379: INTEN Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																												ALM	OF	ACC	ACF

Table 380: 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

8.2.8.2.7 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 381: INTSTAT Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																											ALM	OF	ACC	ACF	

Table 382: 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

8.2.8.2.8 *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 383: INTCLR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																											ALM	OF	ACC	ACF	

Table 384: 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

9. Counter/Timer Module

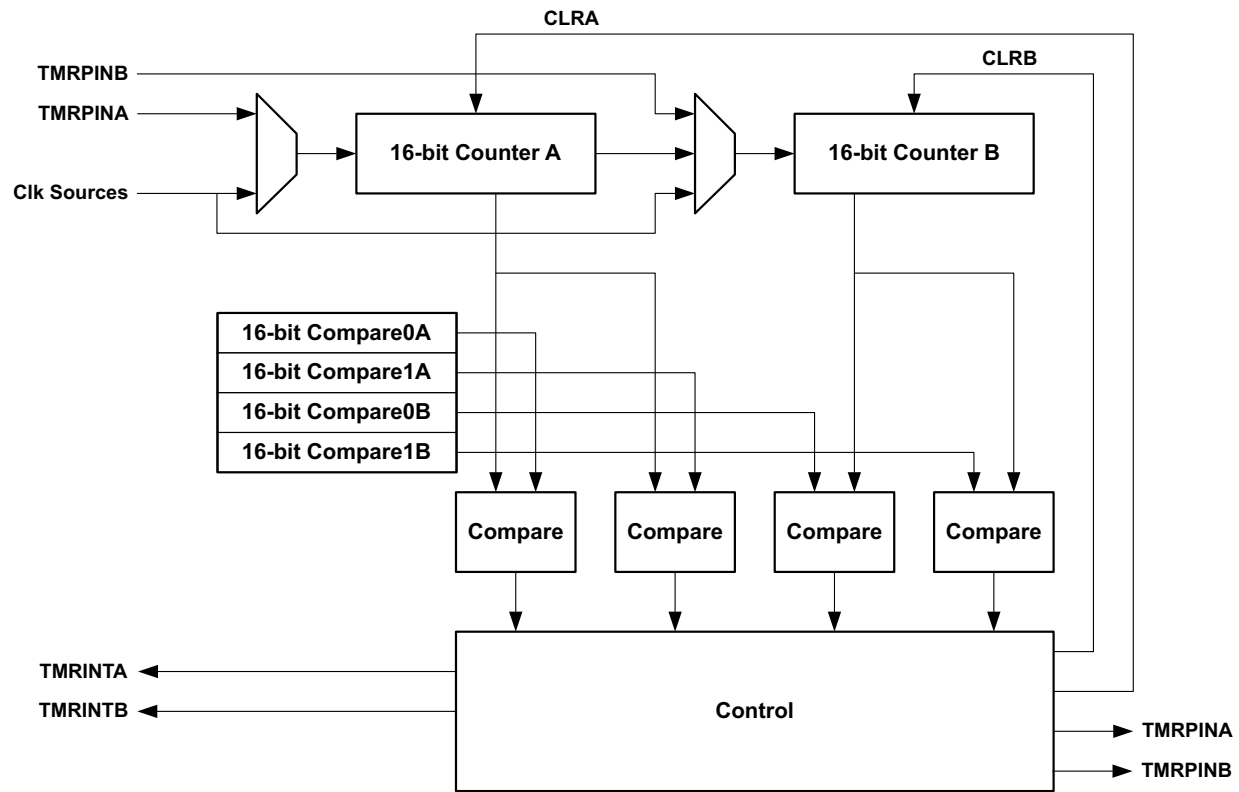


Figure 40. Block Diagram for One Counter/Timer Pair

9.1 Functional Overview

The Apollo MCU Timer/Counter module includes four Timer/Counter pairs, one of which is shown in Figure 40. 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, and can be configured to provide a variety of functions:

- 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

9.2 Counter/Timer Functions

Each Counter/Timer operates in a mode controlled by the REG_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 REG_CTIMER_CMPR0/1 value of zero (a count of 1) is invalid, and that the first measured period will be between the

REG_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.
- 4 => 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 iterations.

9.2.1 Single Count (FN = 0)

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 CLR has been asserted previously. The Timer counts up on each selected clock, and when it reaches the value in the corresponding REG_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.

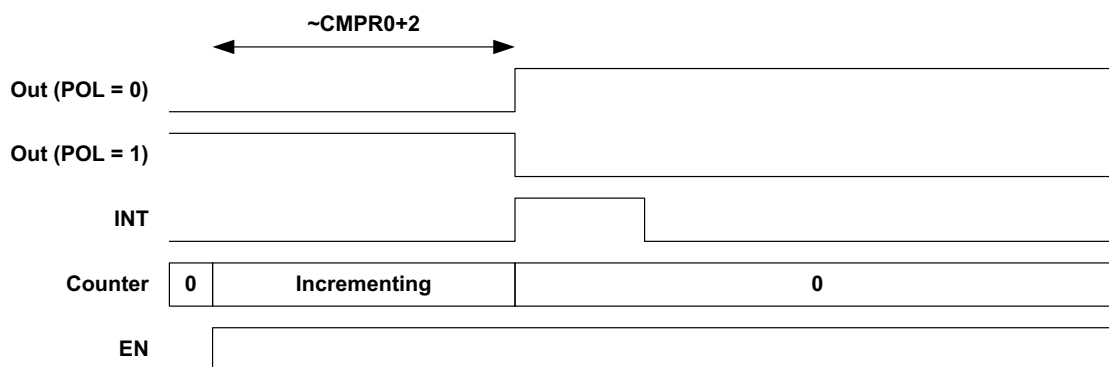


Figure 41. Counter/Timer Operation, FN = 0

9.2.2 Repeated Count (FN = 1)

Operation in this mode is shown in Figure 42. When the Timer is enabled, the pin output is at the level selected by the POL bit and the Timer is at zero because REG_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 REG_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 REG_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.

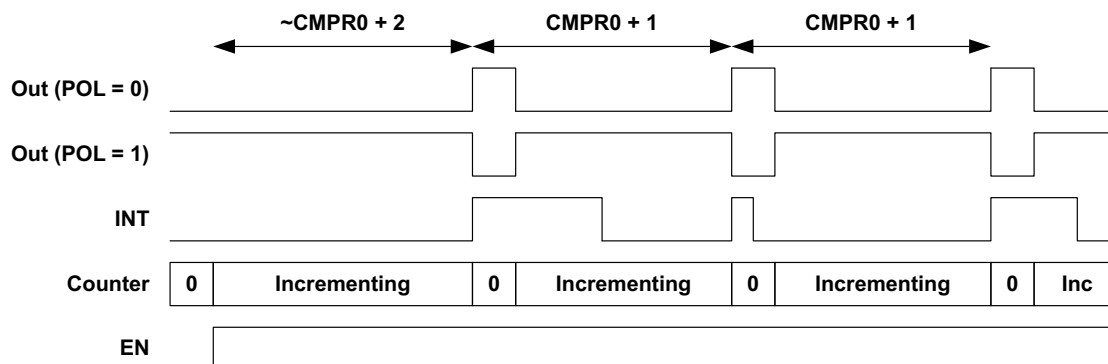


Figure 42. Counter/Timer Operation, FN = 1

9.2.3 Single Pulse (FN = 2)

Operation in this mode is shown in Figure 43. When the Timer is enabled, the pin output is at the level selected by the REG_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 REG_CTIMER_TMRxyPE bit is set) and an interrupt is generated (if the REG_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.

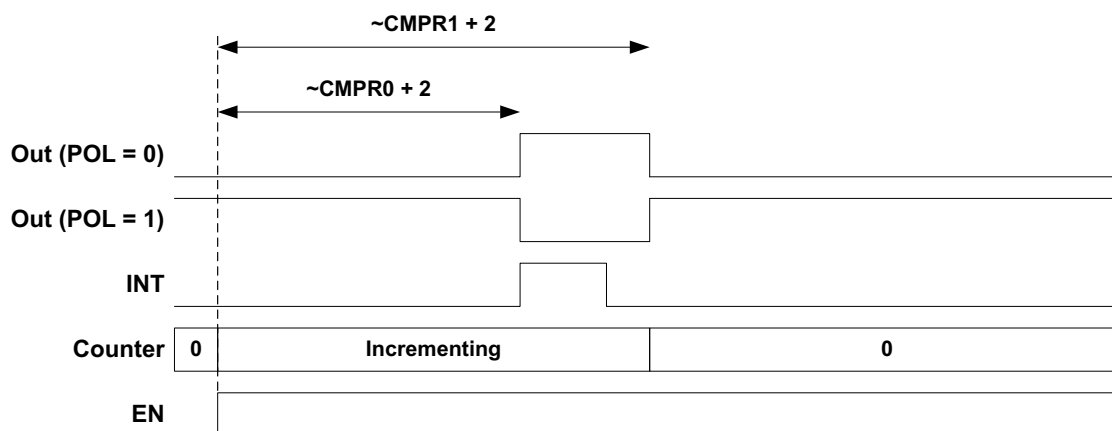


Figure 43. Counter/Timer Operation, FN = 2

9.2.4 Repeated Pulse (FN = 3)

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). 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.

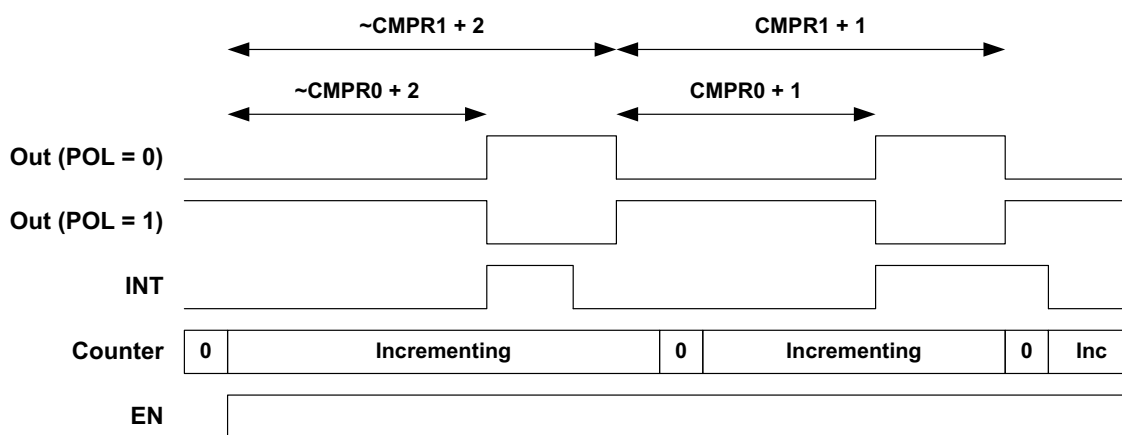


Figure 44. Counter/Timer Operation, FN = 3

9.2.5 Continuous (FN = 4)

Operation in this mode is shown in Figure 45. 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.

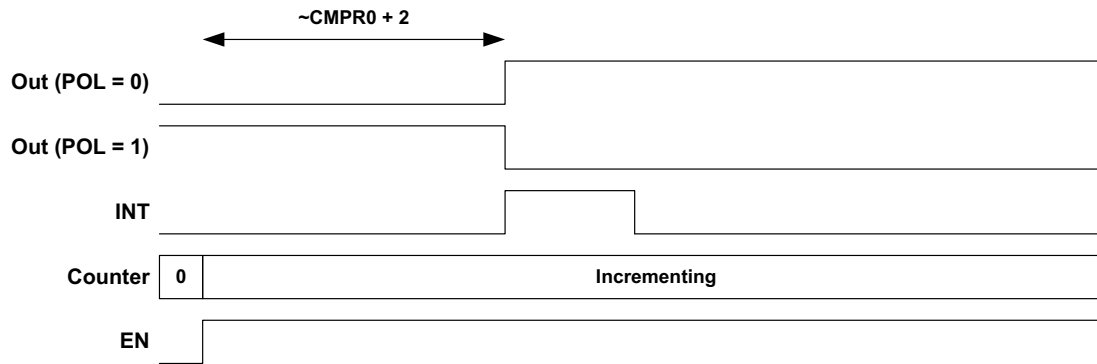


Figure 45. Counter/Timer Operation, FN = 4

9.3 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 REG_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.

9.4 Power Optimization by Measuring HCLK

Each timer has the capability to select the processor clock HCLK as the counter clock input. This allows a very straightforward measurement of how much of the time the processor is in a Sleep or Deep Sleep mode. Two counters are configured with FN = 4 so that they count continuously. One is supplied HCLK as its clock, and the other is supplied with a divided version of the HFRC clock. The two counters are enabled simultaneously, and after some period of system operation they are disabled and read. The HFRC count value defines how much real time has elapsed and how many HCLKs could have occurred in that time, and the HCLK count value defines how many actual HCLKs were received in that time. The ratio is an accurate measurement of the percentage of time the CPU is asleep, and is an effective tool for power optimization.

9.5 Generating the Sample Rate for the ADC

Timer CTTMRA3 has a special function which allows it to be used as the sample trigger generator for the ADC. If the ADCEN and TMRA3PE bits are set, the output of the timer is sent to the ADC which uses it as a trigger. Note that, while the TMRA3PE bit must be set, the TCTA3 GPIO output for the Ctimer should not be enabled to prevent unexpected transitions on the corresponding package pin.

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.

9.6 Measuring Buck Converter Charge Insertion

Each counter may be connected to a pulse stream from one of the two analog Buck Converters. One pulse is generated each time the Buck Converter inserts charge into 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. Two counters are 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. The two counters are enabled simultaneously, and after some period of system operation they are disabled and read. The

LFRC count value defines how much real time has elapsed, and the Buck Converter count value defines how much energy was consumed in that time.

9.7 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.

9.7.1 Register Memory Map

Table 387: 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

9.7.2 CTIMER Registers

9.7.2.1 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 388: TMR0 Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	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
CTTMRB0																CTTMRA0															

Table 389: TMR0 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB0	0x0	RO	Counter/Timer B0.
15:0	CTTMRA0	0x0	RO	Counter/Timer A0.

9.7.2.2 CMPRA0 Register

Counter/Timer A0 Compare Registers

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x40008004

Compare limits for timer half A.

Table 390: CMPRA0 Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	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
CMPR1A0																CMPR0A0															

Table 391: 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 391: 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.

9.7.2.3 CMPRB0 Register

Counter/Timer B0 Compare Registers

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x40008008

Compare limits for timer half B.

Table 392: CMPRB0 Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	0	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	0
CMPR1B0																CMPR0B0																

Table 393: 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.

9.7.2.4 CTRL0 Register

Counter/Timer Control

OFFSET: 0x0000000C**INSTANCE 0 ADDRESS: 0x4000800C**

Control bit fields for both halves of timer 0.

Table 394: CTRL0 Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0					
CTLINK0	RSVD	TMRB0POL	TMRB0CLR	TMRB0PE	TMRB0IE	TMRB0FN					TMRB0CLK					TMRB0EN	RSVD					TMRA0POL	TMRA0CLR	TMRA0PE	TMRA0IE	TMRA0FN					TMRA0CLK					TMRA0EN

Table 395: CTRL0 Register Bits

Bit	Name	Reset	RW	Description
31	CTLINK0	0x0	RW	Counter/Timer A0/B0 Link bit. TWO_16BIT_TIMERS = 0x0 - Use A0/B0 timers as two independent 16-bit timers (default). 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. NORMAL = 0x0 - The polarity of the TMRPINB0 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINB0 pin is the inverse of the timer output.
27	TMRB0CLR	0x0	RW	Counter/Timer B0 Clear bit. RUN = 0x0 - Allow counter/timer B0 to run CLEAR = 0x1 - Holds counter/timer B0 at 0x0000.
26	TMRB0PE	0x0	RW	Counter/Timer B0 Output Enable bit. DIS = 0x0 - Counter/Timer B holds the TMRPINB signal at the value TMRB0POL. EN = 0x1 - Enable counter/timer B0 to generate a signal on TMRPINB.
25	TMRB0IE	0x0	RW	Counter/Timer B0 Interrupt Enable bit. DIS = 0x0 - Disable counter/timer B0 from generating an interrupt. EN = 0x1 - Enable counter/timer B0 to generate an interrupt.
24:22	TMRB0FN	0x0	RW	Counter/Timer B0 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B0, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B0, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B0, assert, count to CMPR1B, deassert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0B0, assert, count to CMPR1B, deassert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.

Table 395: CTRL0 Register Bits

Bit	Name	Reset	RW	Description
21:17	TMRB0CLK	0x0	RW	Counter/Timer B0 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC = 0x1 - Clock source is the HFRC HFRC_DIV8 = 0x2 - Clock source is HFRC / 8 HFRC_DIV128 = 0x3 - Clock source is HFRC / 128 HFRC_DIV512 = 0x4 - Clock source is HFRC / 512 HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC / 16K RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK = 0xF - Clock source is HCLK. BUCKB = 0x10 - Clock source is buck converter stream B.
16	TMRB0EN	0x0	RW	Counter/Timer B0 Enable bit. DIS = 0x0 - Counter/Timer B0 Disable. EN = 0x1 - Counter/Timer B0 Enable.
15:13	RSVD	0x0	RO	RESERVED
12	TMRA0POL	0x0	RW	Counter/Timer A0 output polarity. NORMAL = 0x0 - The polarity of the TMRPINA0 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINA0 pin is the inverse of the timer output.
11	TMRA0CLR	0x0	RW	Counter/Timer A0 Clear bit. RUN = 0x0 - Allow counter/timer A0 to run CLEAR = 0x1 - Holds counter/timer A0 at 0x0000.
10	TMRA0PE	0x0	RW	Counter/Timer A0 Output Enable bit. DIS = 0x0 - Counter/Timer A holds the TMRPINA signal at the value TMRA0POL. EN = 0x1 - Enable counter/timer B0 to generate a signal on TMRPINB.
9	TMRA0IE	0x0	RW	Counter/Timer A0 Interrupt Enable bit. DIS = 0x0 - Disable counter/timer A0 from generating an interrupt. EN = 0x1 - Enable counter/timer A0 to generate an interrupt.
8:6	TMRA0FN	0x0	RW	Counter/Timer A0 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A0, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A0, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A0, assert, count to CMPR1B, deassert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0A0, assert, count to CMPR1A0, deassert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.

Table 395: CTRL0 Register Bits

Bit	Name	Reset	RW	Description
5:1	TMRA0CLK	0x0	RW	Counter/Timer A0 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC = 0x1 - Clock source is the HFRC HFRC_DIV8 = 0x2 - Clock source is HFRC / 8 HFRC_DIV128 = 0x3 - Clock source is HFRC / 128 HFRC_DIV512 = 0x4 - Clock source is HFRC / 512 HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC / 16K RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK = 0xF - Clock source is HCLK. BUCKA = 0x10 - Clock source is buck converter stream A.
0	TMRA0EN	0x0	RW	Counter/Timer A0 Enable bit. DIS = 0x0 - Counter/Timer A0 Disable. EN = 0x1 - Counter/Timer A0 Enable.

9.7.2.5 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 396: TMR1 Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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
CTTMRB1																CTTMRA1															

Table 397: TMR1 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB1	0x0	RO	Counter/Timer B1.
15:0	CTTMRA1	0x0	RO	Counter/Timer A1.

9.7.2.6 CMPRA1 Register

Counter/Timer A1 Compare Registers

OFFSET: 0x00000014

INSTANCE 0 ADDRESS: 0x40008014

This register holds the compare limits for timer half A.

Table 398: CMPRA1 Register

3	3	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
CMPR1A1																CMPR0A1															

Table 399: 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.

9.7.2.7 CMPRB1 Register

Counter/Timer B1 Compare Registers

OFFSET: 0x00000018

INSTANCE 0 ADDRESS: 0x40008018

This register holds the compare limits for timer half B.

Table 400: CMPRB1 Register

3	3	2	2	2	2	2	2	2	2	2	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
CMPR1B1																CMPR0B1															

Table 401: 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.

9.7.2.8 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 402: CTRL1 Register

3	3	2	2	2	2	2	2	2	2	2	2	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					
CTLINK1	RSVD	TMRB1POL	TMRB1CLR	TMRB1PE	TMRB1IE	TMRB1FN					TMRB1CLK					TMRB1EN	RSVD					TMRA1POL	TMRA1CLR	TMRA1PE	TMRA1IE	TMRA1FN					TMRA1CLK					TMRA1EN

Table 403: CTRL1 Register Bits

Bit	Name	Reset	RW	Description
31	CTLINK1	0x0	RW	Counter/Timer A1/B1 Link bit. TWO_16BIT_TIMERS = 0x0 - Use A0/B0 timers as two independent 16-bit timers (default). 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. NORMAL = 0x0 - The polarity of the TMRPINB1 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINB1 pin is the inverse of the timer output.
27	TMRB1CLR	0x0	RW	Counter/Timer B1 Clear bit. RUN = 0x0 - Allow counter/timer B1 to run CLEAR = 0x1 - Holds counter/timer B1 at 0x0000.
26	TMRB1PE	0x0	RW	Counter/Timer B1 Output Enable bit. DIS = 0x0 - Counter/Timer B holds the TMRPINB signal at the value TMRB1POL. EN = 0x1 - Enable counter/timer B1 to generate a signal on TMRPINB.
25	TMRB1IE	0x0	RW	Counter/Timer B1 Interrupt Enable bit. DIS = 0x0 - Disable counter/timer B1 from generating an interrupt. EN = 0x1 - Enable counter/timer B1 to generate an interrupt.

Table 403: CTRL1 Register Bits

Bit	Name	Reset	RW	Description
24:22	TMRB1FN	0x0	RW	Counter/Timer B1 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B1, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B1, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B1, assert, count to CMPR1B, deassert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0B1, assert, count to CMPR1B1, deassert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.
21:17	TMRB1CLK	0x0	RW	Counter/Timer B1 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC = 0x1 - Clock source is the HFRC HFRC_DIV8 = 0x2 - Clock source is HFRC / 8 HFRC_DIV128 = 0x3 - Clock source is HFRC / 128 HFRC_DIV512 = 0x4 - Clock source is HFRC / 512 HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC / 16K RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK = 0xF - Clock source is HCLK. BUCKB = 0x10 - Clock source is buck converter stream B.
16	TMRB1EN	0x0	RW	Counter/Timer B1 Enable bit. DIS = 0x0 - Counter/Timer B1 Disable. EN = 0x1 - Counter/Timer B1 Enable.
15:13	RSVD	0x0	RO	RESERVED
12	TMRA1POL	0x0	RW	Counter/Timer A1 output polarity. NORMAL = 0x0 - The polarity of the TMRPINA1 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINA1 pin is the inverse of the timer output.
11	TMRA1CLR	0x0	RW	Counter/Timer A1 Clear bit. RUN = 0x0 - Allow counter/timer A1 to run CLEAR = 0x1 - Holds counter/timer A1 at 0x0000.
10	TMRA1PE	0x0	RW	Counter/Timer A1 Output Enable bit. DIS = 0x0 - Counter/Timer A holds the TMRPINA signal at the value TMRA1POL. EN = 0x1 - Enable counter/timer A1 to generate a signal on TMRPINA.
9	TMRA1IE	0x0	RW	Counter/Timer A1 Interrupt Enable bit. DIS = 0x0 - Disable counter/timer A1 from generating an interrupt. EN = 0x1 - Enable counter/timer A1 to generate an interrupt.

Table 403: CTRL1 Register Bits

Bit	Name	Reset	RW	Description
8:6	TMRA1FN	0x0	RW	Counter/Timer A1 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A1, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A1, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A1, assert, count to CMPR1B, deassert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0A1, assert, count to CMPR1A1, deassert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.
5:1	TMRA1CLK	0x0	RW	Counter/Timer A1 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC = 0x1 - Clock source is the HFRC HFRC_DIV8 = 0x2 - Clock source is the HFRC / 8 HFRC_DIV128 = 0x3 - Clock source is HFRC / 128 HFRC_DIV512 = 0x4 - Clock source is HFRC / 512 HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC / 16K RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK = 0xF - Clock source is HCLK. BUCKA = 0x10 - Clock source is buck converter stream A.
0	TMRA1EN	0x0	RW	Counter/Timer A1 Enable bit. DIS = 0x0 - Counter/Timer A1 Disable. EN = 0x1 - Counter/Timer A1 Enable.

9.7.2.9 TMR2 Register

Counter/Timer Register

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x40008020

Counter/Timer Register

Table 404: TMR2 Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	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
CTTMRB2															CTTMRA2												

Bit	Name	Reset	RW	Description
31:16	CTTMRB2	0x0	RO	Counter/Timer B2.
15:0	CTTMRA2	0x0	RO	Counter/Timer A2.

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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
CMPR1A2																CMPR0A2															

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.

3	3	2	2	2	2	2	2	2	2	2	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
CMPR1B2															CMPR0B2																

Table 409: 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.

9.7.2.12 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 410: CTRL2 Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	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
CTLINK2	RSVD	TMRB2POL	TMRB2CLR	TMRB2PE	TMRB2IE	TMRB2FN	TMRB2CLK					TMRB2EN	RSVD	TMRA2POL	TMRA2CLR	TMRA2PE	TMRA2IE	TMRA2FN	TMRA2CLK					TMRA2EN							

Table 411: CTRL2 Register Bits

Bit	Name	Reset	RW	Description
31	CTLINK2	0x0	RW	Counter/Timer A2/B2 Link bit. TWO_16BIT_TIMERS = 0x0 - Use A0/B0 timers as two independent 16-bit timers (default). 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. NORMAL = 0x0 - The polarity of the TMRPINB2 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINB2 pin is the inverse of the timer output.
27	TMRB2CLR	0x0	RW	Counter/Timer B2 Clear bit. RUN = 0x0 - Allow counter/timer B2 to run CLEAR = 0x1 - Holds counter/timer B2 at 0x0000.

Table 411: CTRL2 Register Bits

Bit	Name	Reset	RW	Description
26	TMRB2PE	0x0	RW	Counter/Timer B2 Output Enable bit. DIS = 0x0 - Counter/Timer B holds the TMRPINB signal at the value TMRB2POL. EN = 0x1 - Enable counter/timer B2 to generate a signal on TMRPINB.
25	TMRB2IE	0x0	RW	Counter/Timer B2 Interrupt Enable bit. DIS = 0x0 - Disable counter/timer B2 from generating an interrupt. EN = 0x1 - Enable counter/timer B2 to generate an interrupt.
24:22	TMRB2FN	0x0	RW	Counter/Timer B2 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B2, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B2, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B2, assert, count to CMPR1B, deassert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0B2, assert, count to CMPR1B2, deassert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.
21:17	TMRB2CLK	0x0	RW	Counter/Timer B2 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC = 0x1 - Clock source is the HFRC HFRC_DIV8 = 0x2 - Clock source is HFRC / 8 HFRC_DIV128 = 0x3 - Clock source is HFRC / 128 HFRC_DIV512 = 0x4 - Clock source is HFRC / 512 HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC / 16K RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK = 0xF - Clock source is HCLK. BUCKA = 0x10 - Clock source is buck converter stream A.
16	TMRB2EN	0x0	RW	Counter/Timer B2 Enable bit. DIS = 0x0 - Counter/Timer B2 Disable. EN = 0x1 - Counter/Timer B2 Enable.
15:13	RSVD	0x0	RO	RESERVED
12	TMRA2POL	0x0	RW	Counter/Timer A2 output polarity. NORMAL = 0x0 - The polarity of the TMRPINA2 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINA2 pin is the inverse of the timer output.
11	TMRA2CLR	0x0	RW	Counter/Timer A2 Clear bit. RUN = 0x0 - Allow counter/timer A2 to run CLEAR = 0x1 - Holds counter/timer A2 at 0x0000.

Table 411: CTRL2 Register Bits

Bit	Name	Reset	RW	Description
10	TMRA2PE	0x0	RW	Counter/Timer A2 Output Enable bit. DIS = 0x0 - Counter/Timer A holds the TMRPINA signal at the value TMRA2POL. EN = 0x1 - Enable counter/timer A2 to generate a signal on TMRPINA.
9	TMRA2IE	0x0	RW	Counter/Timer A2 Interrupt Enable bit. DIS = 0x0 - Disable counter/timer A2 from generating an interrupt. EN = 0x1 - Enable counter/timer A2 to generate an interrupt.
8:6	TMRA2FN	0x0	RW	Counter/Timer A2 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A2, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A2, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A2, assert, count to CMPR1B, deassert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0A2, assert, count to CMPR1A2, deassert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.
5:1	TMRA2CLK	0x0	RW	Counter/Timer A2 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC = 0x1 - Clock source is the HFRC HFRC_DIV8 = 0x2 - Clock source is HFRC / 8 HFRC_DIV128 = 0x3 - Clock source is HFRC / 128 HFRC_DIV512 = 0x4 - Clock source is HFRC / 512 HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC / 16K RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK = 0xF - Clock source is HCLK. BUCKB = 0x10 - Clock source is buck converter stream B.
0	TMRA2EN	0x0	RW	Counter/Timer A2 Enable bit. DIS = 0x0 - Counter/Timer A2 Disable. EN = 0x1 - Counter/Timer A2 Enable.

9.7.2.13 TMR3 Register

Counter/Timer Register

OFFSET: 0x00000030

INSTANCE 0 ADDRESS: 0x40008030

Counter/Timer Register

Table 412: TMR3 Register

3	3	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	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
CTTMRB3																CTTMRA3															

Table 413: TMR3 Register Bits

Bit	Name	Reset	RW	Description
31:16	CTTMRB3	0x0	RO	Counter/Timer B3.
15:0	CTTMRA3	0x0	RO	Counter/Timer A3.

9.7.2.14 CMPRA3 Register

Counter/Timer A3 Compare Registers

OFFSET: 0x00000034

INSTANCE 0 ADDRESS: 0x40008034

This register holds the compare limits for timer half A.

Table 414: CMPRA3 Register

3	3	2	2	2	2	2	2	2	2	2	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
CMPR1A3																CMPR0A3															

Table 415: 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.

9.7.2.15 CMPRB3 Register

Counter/Timer B3 Compare Registers

OFFSET: 0x00000038**INSTANCE 0 ADDRESS: 0x40008038**

This register holds the compare limits for timer half B.

Table 416: CMPRB3 Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	0	0	0	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
CMPR1B3																CMPR0B3															

Table 417: 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.

9.7.2.16 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 418: CTRL3 Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
CTLINK3	RSVD	TMRB3POL	TMRB3CLR	TMRB3PE	TMRB3IE	TMRB3FN				TMRB3CLK				TMRB3EN	ADCEN	RSVD	TMRA3POL	TMRA3CLR	TMRA3PE	TMRA3IE	TMRA3FN				TMRA3CLK				TMRA3EN		

Table 419: CTRL3 Register Bits

Bit	Name	Reset	RW	Description
31	CTLINK3	0x0	RW	Counter/Timer A/B Link bit. TWO_16BIT_TIMERS = 0x0 - Use A0/B0 timers as two independent 16-bit timers (default). 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. NORMAL = 0x0 - The polarity of the TMRPINB3 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINB3 pin is the inverse of the timer output.
27	TMRB3CLR	0x0	RW	Counter/Timer B3 Clear bit. RUN = 0x0 - Allow counter/timer B3 to run. CLEAR = 0x1 - Holds counter/timer B3 at 0x0000.
26	TMRB3PE	0x0	RW	Counter/Timer B3 Output Enable bit. DIS = 0x0 - Counter/Timer B holds the TMRPINB signal at the value TMRB3POL. EN = 0x1 - Enable counter/timer B3 to generate a signal on TMRPINB.
25	TMRB3IE	0x0	RW	Counter/Timer B3 Interrupt Enable bit. DIS = 0x0 - Disable counter/timer B3 from generating an interrupt. EN = 0x1 - Enable counter/timer B3 to generate an interrupt.
24:22	TMRB3FN	0x0	RW	Counter/Timer B3 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0B3, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0B3, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0B3, assert, count to CMPR1B, deassert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0B3, assert, count to CMPR1B3, deassert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.

Table 419: CTRL3 Register Bits

Bit	Name	Reset	RW	Description
21:17	TMRB3CLK	0x0	RW	Counter/Timer B3 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINB. HFRC = 0x1 - Clock source is the HFRC HFRC_DIV8 = 0x2 - Clock source is HFRC / 8 HFRC_DIV128 = 0x3 - Clock source is HFRC / 128 HFRC_DIV512 = 0x4 - Clock source is HFRC / 512 HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC / 16K RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK = 0xF - Clock source is HCLK. BUCKA = 0x10 - Clock source is buck converter stream A.
16	TMRB3EN	0x0	RW	Counter/Timer B3 Enable bit. DIS = 0x0 - Counter/Timer B3 Disable. 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. NORMAL = 0x0 - The polarity of the TMRPINA3 pin is the same as the timer output. INVERTED = 0x1 - The polarity of the TMRPINA3 pin is the inverse of the timer output.
11	TMRA3CLR	0x0	RW	Counter/Timer A3 Clear bit. CLEAR = 0x1 - Holds counter/timer A3 at 0x0000.
10	TMRA3PE	0x0	RW	Counter/Timer A3 Output Enable bit. DIS = 0x0 - Counter/Timer A holds the TMRPINA signal at the value TMRA3POL. EN = 0x1 - Enable counter/timer A3 to generate a signal on TMRPINA.
9	TMRA3IE	0x0	RW	Counter/Timer A3 Interrupt Enable bit. DIS = 0x0 - Disable counter/timer A3 from generating an interrupt. EN = 0x1 - Enable counter/timer A3 to generate an interrupt.

Table 419: CTRL3 Register Bits

Bit	Name	Reset	RW	Description
8:6	TMRA3FN	0x0	RW	Counter/Timer A3 Function Select. SINGLECOUNT = 0x0 - Single count (output toggles and sticks). Count to CMPR0A3, stop. REPEATEDCOUNT = 0x1 - Repeated count (periodic 1-clock-cycle-wide pulses). Count to CMPR0A3, restart. PULSE_ONCE = 0x2 - Pulse once (aka one-shot). Count to CMPR0A3, assert, count to CMPR1B, deassert, stop. PULSE_CONT = 0x3 - Pulse continuously. Count to CMPR0A3, assert, count to CMPR1A3, deassert, restart. CONTINUOUS = 0x4 - Continuous run (aka Free Run). Count continuously.
5:1	TMRA3CLK	0x0	RW	Counter/Timer A3 Clock Select. TMRPIN = 0x0 - Clock source is TMRPINA. HFRC = 0x1 - Clock source is the HFRC HFRC_DIV8 = 0x2 - Clock source is HFRC / 8 HFRC_DIV128 = 0x3 - Clock source is HFRC / 128 HFRC_DIV512 = 0x4 - Clock source is HFRC / 512 HFRC_DIV2K = 0x5 - Clock source is HFRC / 2048 XT = 0x6 - Clock source is the XT (uncalibrated). XT_DIV2 = 0x7 - Clock source is XT / 2 XT_DIV16 = 0x8 - Clock source is XT / 16 XT_DIV256 = 0x9 - Clock source is XT / 256 LFRC_DIV2 = 0xA - Clock source is LFRC / 2 LFRC_DIV32 = 0xB - Clock source is LFRC / 32 LFRC_DIV1K = 0xC - Clock source is LFRC / 1024 LFRC = 0xD - Clock source is LFRC / 16K RTC_100HZ = 0xE - Clock source is 100 Hz from the current RTC oscillator. HCLK = 0xF - Clock source is HCLK. BUCKB = 0x10 - Clock source is buck converter stream B.
0	TMRA3EN	0x0	RW	Counter/Timer A3 Enable bit. DIS = 0x0 - Counter/Timer A3 Disable. EN = 0x1 - Counter/Timer A3 Enable.

9.7.2.17 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 420: INTEN Register

3	3	2	2	2	2	2	2	2	2	2	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
RSVD																					CTMRB3INT	CTMRA3INT	CTMRB2INT	CTMRA2INT	CTMRB1INT	CTMRA1INT	CTMRB0INT	CTMRA0INT

Table 421: 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.

9.7.2.18 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 422: INTSTAT Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																								CTMRB3INT	CTMRA3INT	CTMRB2INT	CTMRA2INT	CTMRB1INT	CTMRA1INT	CTMRB0INT	CTMRA0INT

Table 423: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:8	RSVD	0x0	RO	RESERVED
7	CTMRB3INT	0x0	RW	Counter/Timer B3 interrupt.

Table 423: 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.

9.7.2.19 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 424: INTCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSVD																								CTMRB3INT	CTMRA3INT	CTMRB2INT	CTMRA2INT	CTMRB1INT	CTMRA1INT	CTMRB0INT	CTMRA0INT

Table 425: 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 425: 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.

9.7.2.20 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 426: INTSET Register

3	3	2	2	2	2	2	2	2	2	2	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
RSVD																					CTMRB3INT	CTMRA3INT	CTMRB2INT	CTMRA2INT	CTMRB1INT	CTMRA1INT	CTMRB0INT	CTMRA0INT

Table 427: 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 427: 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.

10. Watchdog Timer Module

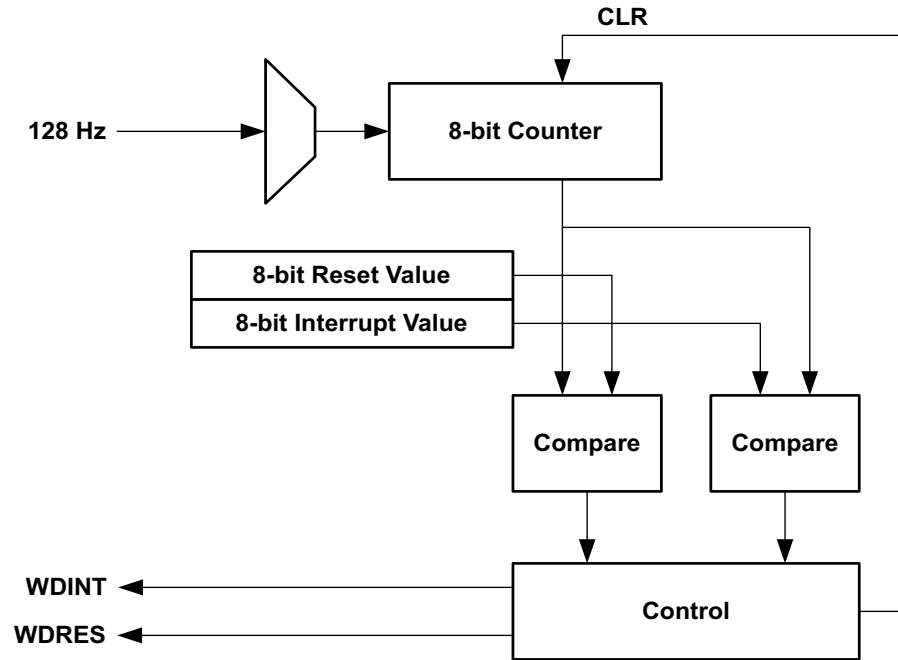


Figure 46. Block diagram for the Watchdog Timer Module

10.1 Functional Overview

The Watchdog Timer (WDT), shown in Figure 46, is used to insure that software is operational, by resetting the Apollo MCU if the WDT reaches a configurable value before being cleared by software. The WDT is 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.

10.2 WDT Registers

Watchdog Timer

INSTANCE 0 BASE ADDRESS:0x40024000

10.2.1 Register Memory Map

Table 428: 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

10.2.2 WDT Registers

10.2.2.1 CFG Register

Configuration Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x40024000

Configuration Register

Table 429: CFG Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD								INTVAL								RESVAL								RSVD					RESEN	INTEN	WDTEN

Table 430: 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 watch-dog interrupt.

Bit	Name	Reset	RW	Description
15:8	RESVAL	0x0	RW	This bitfield is the compare value for counter bits 7:0 to generate a watchdog reset.
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.

3	3	2	2	2	2	2	2	2	2	2	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								
RSVD																						RSTRT															

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. KEYVALUE = 0xB2 - This is the key value to write to WDTRSTRT to restart the WDT.

10.2.2.3 LOCK Register

Locks the WDT

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x40024008

Locks the WDT

Table 433: LOCK Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	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
RSVD																					LOCK										

Table 434: 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. KEYVALUE = 0x3A - This is the key value to write to WDTLOCK to lock the WDT.

10.2.2.4 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 435: INTEN Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																															WDT

Table 436: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	This bitfield is reserved for future use.

Table 436: INTEN Register Bits

Bit	Name	Reset	RW	Description
0	WDT	0x0	RW	Watchdog Timer Interrupt.

10.2.2.5 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 437: INTSTAT Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																															WDT

Table 438: 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.

10.2.2.6 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 439: INTCLR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																														WDT	

Table 440: 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.

10.2.2.7 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 441: INTSET Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																															WDT

Table 442: 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.

11. Reset Generator Module

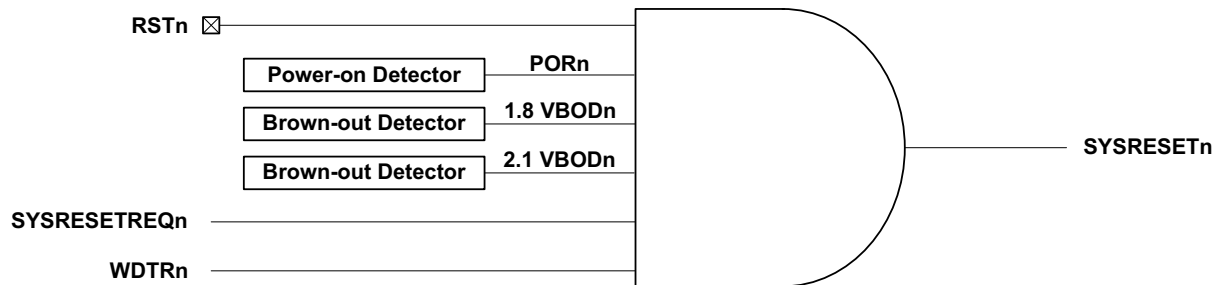


Figure 47. Block diagram for the Reset Generator Module

11.1 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 RSTSTAT register described in Section RSTGEN Registers.

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

11.2 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 debounce circuit ensures that glitches on RSTn do 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 48

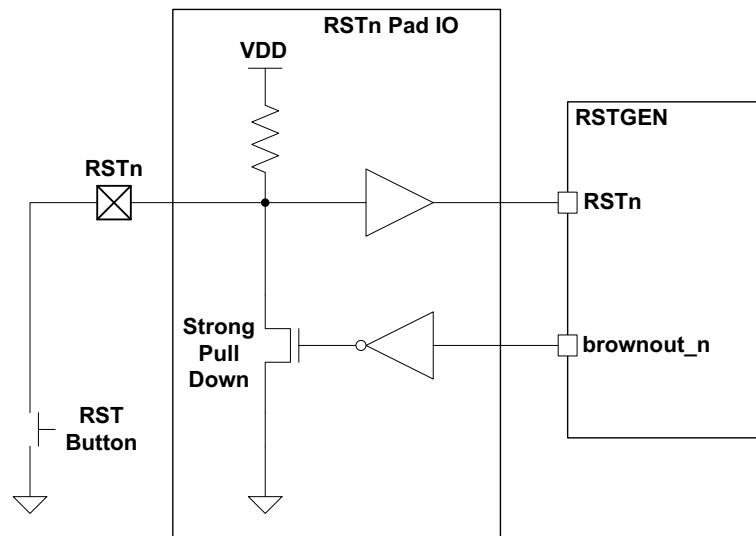


Figure 48. Block diagram of circuitry for Reset pin

11.3 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.

11.4 Brown-out Event

An integrated brown-out detector monitors the supply voltage and signals when the voltage has fallen below the 2.1 V and 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, the reset module will initiate a SYSRESETn and the 1.8VBODn, is not maskable. A brown-out event will enable the RSTn pull-down and drive the reset pin low.

11.5 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.

11.6 Watchdog Expiration

The Timers and Counters Module described in Section 10.1 includes a watchdog timer sub-module that generates an interrupt if it has not been properly managed by software within a pre-defined time. The watchdog reset is maskable.

Table 445: 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.

11.7.3.1 SWPOI Register

Software POI Reset

OFFSET: 0x00000004

INSTANCE 0 ADDRESS: 0x40000004

Software POI Reset

Table 446: SWPOI Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	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								
RSVD																						SWPOIKEY													

Table 447: 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. KEYVALUE = 0x1B - Writing 0x1B key value generates a software POI reset.

11.7.3.2 SWPOR Register

Software POR Reset

OFFSET: 0x00000008

INSTANCE 0 ADDRESS: 0x40000008

Software POR Reset

Table 448: SWPOR Register

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	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
RSVD																					SWPORKEY										

Table 449: 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. KEYVALUE = 0xD4 - Writing 0xD4 key value generates a software POR reset.

11.7.3.3 STAT Register

Status Register

OFFSET: 0x0000000C

INSTANCE 0 ADDRESS: 0x4000000C

Status Register

Table 450: STAT Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																									WDRSTAT	DBGRSTAT	POIRSTAT	SWRSTAT	BORSTAT	PORSTAT	EXRSTAT

Table 451: STAT Register Bits

Bit	Name	Reset	RW	Description
31:7	RSVD	0x0	RO	RESERVED.

Table 451: 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.

11.7.3.4 CLRSTAT Register

Clear the status register

OFFSET: 0x00000010

INSTANCE 0 ADDRESS: 0x40000010

Clear the status register

Table 452: CLRSTAT Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																															CLRSTAT

Table 453: 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.

11.7.3.5 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 454: INTEN Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																															BODH

Table 455: 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.

11.7.3.6 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 456: INTSTAT Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																															BODH

Table 457: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:1	RSVD	0x0	RO	RESERVED.

Table 457: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
0	BODH	0x0	RW	Enables an interrupt that triggers when VCC is below BODH level.

11.7.3.7 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 458: INTCLR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
RSVD																																BODH

Table 459: 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.

11.7.3.8 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 460: INTSET Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																														BODH	

Table 461: 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.

12. UART Module

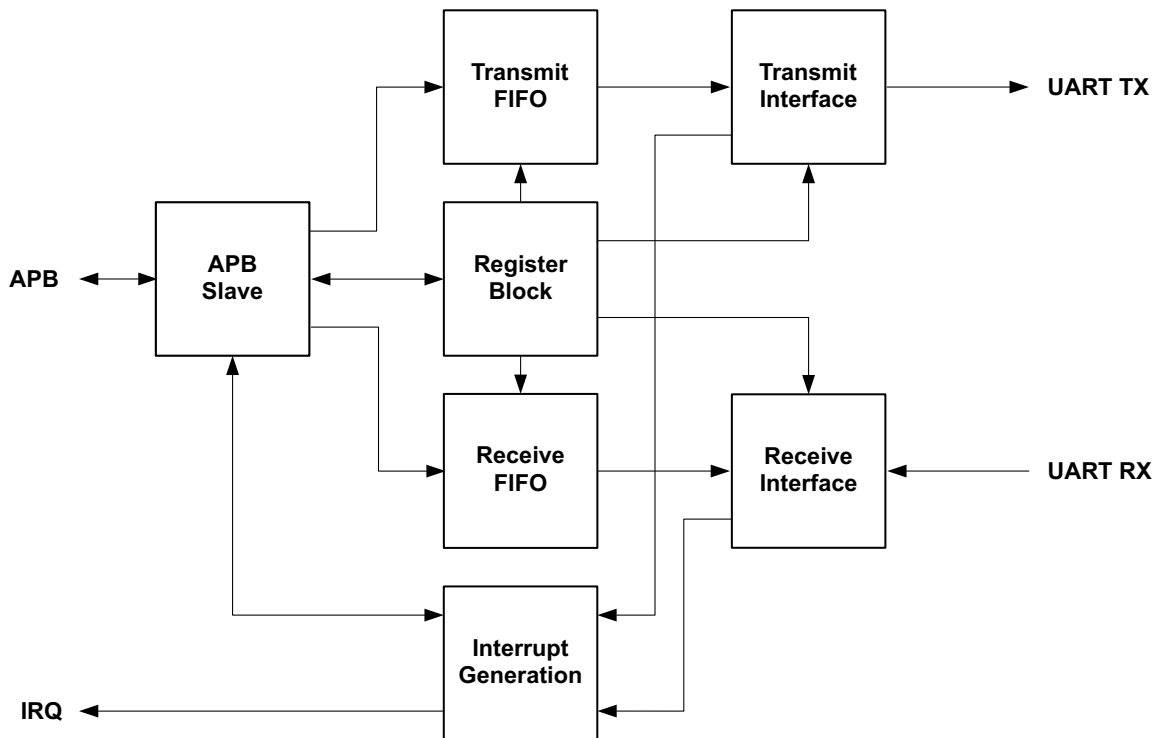


Figure 49. Block Diagram for the UART Module

12.1 Features

The UART Module includes the following key features:

- Operates independently, allowing the MCU 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 capable of a maximum rate of 921,600 bits per second
- Fully programmable data size, parity, and stop bit length
- Programmable hardware flow control
- Support for full-duplex and half-duplex communication
- Loopback functionality for diagnostics and testing

12.2 Functional Overview

Shown in Figure 49, 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 which is capable of operating at a maximum of 921,600 bits per second. 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.

12.3 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.

12.4 Configuration

The UART Register Block in Figure 49 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.

12.5 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.

12.6 UART Registers

Serial UART

INSTANCE 0 BASE ADDRESS:0x4001C000

12.6.1 Register Memory Map

Table 462: 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

12.6.2 UART Registers

12.6.2.1 DR Register

UART Data Register

OFFSET: 0x00000000

INSTANCE 0 ADDRESS: 0x4001C000

UART Data Register

Table 463: DR Register

3	3	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						
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				
RSVD																				OEDATA	BEDATA	PEDATA	FEDATA	DATA											

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. NOERR = 0x0 - No error on UART OEDATA, overrun error indicator. ERR = 0x1 - Error on UART OEDATA, overrun error indicator.
10	BEDATA	0x0	RO	This is the break error indicator. NOERR = 0x0 - No error on UART BEDATA, break error indicator. ERR = 0x1 - Error on UART BEDATA, break error indicator.
9	PEDATA	0x0	RO	This is the parity error indicator. NOERR = 0x0 - No error on UART PEDATA, parity error indicator. ERR = 0x1 - Error on UART PEDATA, parity error indicator.
8	FEDATA	0x0	RO	This is the framing error indicator. NOERR = 0x0 - No error on UART FEDATA, framing error indicator. ERR = 0x1 - Error on UART FEDATA, framing error indicator.
7:0	DATA	0x0	RW	This is the UART data port.

UART Status Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																												OESTAT	BESTAT	PESTAT	FESTAT

Bit	Name	Reset	RW	Description
31:4	RSVD	0x0	RO	This bitfield is reserved for future use.

Table 466: RSR Register Bits

Bit	Name	Reset	RW	Description
3	OESTAT	0x0	RW	This is the overrun error indicator. NOERR = 0x0 - No error on UART OESTAT, overrun error indicator. ERR = 0x1 - Error on UART OESTAT, overrun error indicator.
2	BESTAT	0x0	RW	This is the break error indicator. NOERR = 0x0 - No error on UART BESTAT, break error indicator. ERR = 0x1 - Error on UART BESTAT, break error indicator.
1	PESTAT	0x0	RW	This is the parity error indicator. NOERR = 0x0 - No error on UART PESTAT, parity error indicator. ERR = 0x1 - Error on UART PESTAT, parity error indicator.
0	FESTAT	0x0	RW	This is the framing error indicator. NOERR = 0x0 - No error on UART FESTAT, framing error indicator. ERR = 0x1 - Error on UART FESTAT, framing error indicator.

12.6.2.3 FR Register

Flag Register

OFFSET: 0x00000018

INSTANCE 0 ADDRESS: 0x4001C018

Flag Register

Table 467: FR Register

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	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
RSVD																					RI	TXFE	RXFF	TXFF	RXFE	BUSY	DCD	DSR	CTS		

Table 468: 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. XMTFIFO_EMPTY = 0x1 - Transmit fifo is empty.
6	RXFF	0x0	RO	This bit holds the receive FIFO full indicator. RCVFIFO_FULL = 0x1 - Receive fifo is full.

Bit	Name	Reset	RW	Description
5	TXFF	0x0	RO	This bit holds the transmit FIFO full indicator. XMTFIFO_FULL = 0x1 - Transmit fifo is full.
4	RXFE	0x0	RO	This bit holds the receive FIFO empty indicator. RCVFIFO_EMPTY = 0x1 - Receive fifo is empty.
3	BUSY	0x0	RO	This bit holds the busy indicator. BUSY = 0x1 - UART busy indicator.
2	DCD	0x0	RO	This bit holds the data carrier detect indicator. DETECTED = 0x1 - Data carrier detect detected.
1	DSR	0x0	RO	This bit holds the data set ready indicator. READY = 0x1 - Data set ready.
0	CTS	0x0	RO	This bit holds the clear to send indicator. CLEARTOSEND = 0x1 - Clear to send is indicated.

IrDA Counter

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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						
RSVD																						ILPDVSR															

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.

12.6.2.7 LCRH Register

Line Control High

OFFSET: 0x0000002C

INSTANCE 0 ADDRESS: 0x4001C02C

Line Control High

Table 475: LCRH Register

3	3	2	2	2	2	2	2	2	2	2	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
RSVD																					SPS	WLEN		FEN	STP2	EPS	PEN	BRK

Table 476: 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.

12.6.2.8 CR Register

Control Register

OFFSET: 0x00000030

INSTANCE 0 ADDRESS: 0x4001C030

Control Register

Table 477: CR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																CTSEN	RTSEN	OUT2	OUT1	RTS	DTR	RXE	TXE	LBE	CLKSEL			CLKEN	SIRLP	SIREN	UARTEN

Table 478: 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 478: CR Register Bits

Bit	Name	Reset	RW	Description
6:4	CLKSEL	0x0	RW	This bitfield is the UART clock select. NOCLK = 0x0 - No UART clock. This is the low power default. 24MHZ = 0x1 - 24 MHz clock. Must be used if CLKGEN CORESEL=0. 12MHZ = 0x2 - 12 MHz clock. Must be used if CLKGEN CORESEL=1. Note that CORESEL=1 is unsupported by the IO Master. 6MHZ = 0x3 - 6 MHz clock. Must be used if CLKGEN CORESEL=2, 3, or 4. Note that CORESEL=2 is unsupported. 3MHZ = 0x4 - 3 MHz clock. Must be used if CLKGEN CORESEL=5, 6, or 7. RSVD5 = 0x5 - Reserved. RSVD6 = 0x6 - Reserved. 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.

12.6.2.9 IFLS Register

FIFO Interrupt Level Select

OFFSET: 0x00000034

INSTANCE 0 ADDRESS: 0x4001C034

FIFO Interrupt Level Select

Table 479: IFLS Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																									RXIFLSEL		TXIFLSEL				

Table 480: 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.

12.6.2.10 IER Register

Interrupt Enable

OFFSET: 0x00000038

INSTANCE 0 ADDRESS: 0x4001C038

Interrupt Enable

Table 481: IER Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																					OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM	DSRMIM	DCDMIM	CTSMIM	RIMIM

Table 482: 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.

12.6.2.12 MIS Register

Masked Interrupt Status

OFFSET: 0x00000040

INSTANCE 0 ADDRESS: 0x4001C040

Masked Interrupt Status

Table 485: MIS Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																					OEMIS	BEMIS	PEMIS	FEMIS	RTMIS	TXMIS	RXMIS	DSRMMIS	DCDMMIS	CTSMMS	RIMMIS

Table 486: 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	CTSMIS	0x0	RO	This bit holds the modem CTS interrupt status masked.
0	RIMIS	0x0	RO	This bit holds the modem RI interrupt status masked.

12.6.2.13 IEC Register

Interrupt Clear

OFFSET: 0x00000044

INSTANCE 0 ADDRESS: 0x4001C044

Interrupt Clear

Table 487: IEC Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																					OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC	DSRMIC	DCDMIC	CTSMIC	RIMIC

Table 488: 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.

13. ADC and Temperature Sensor Module

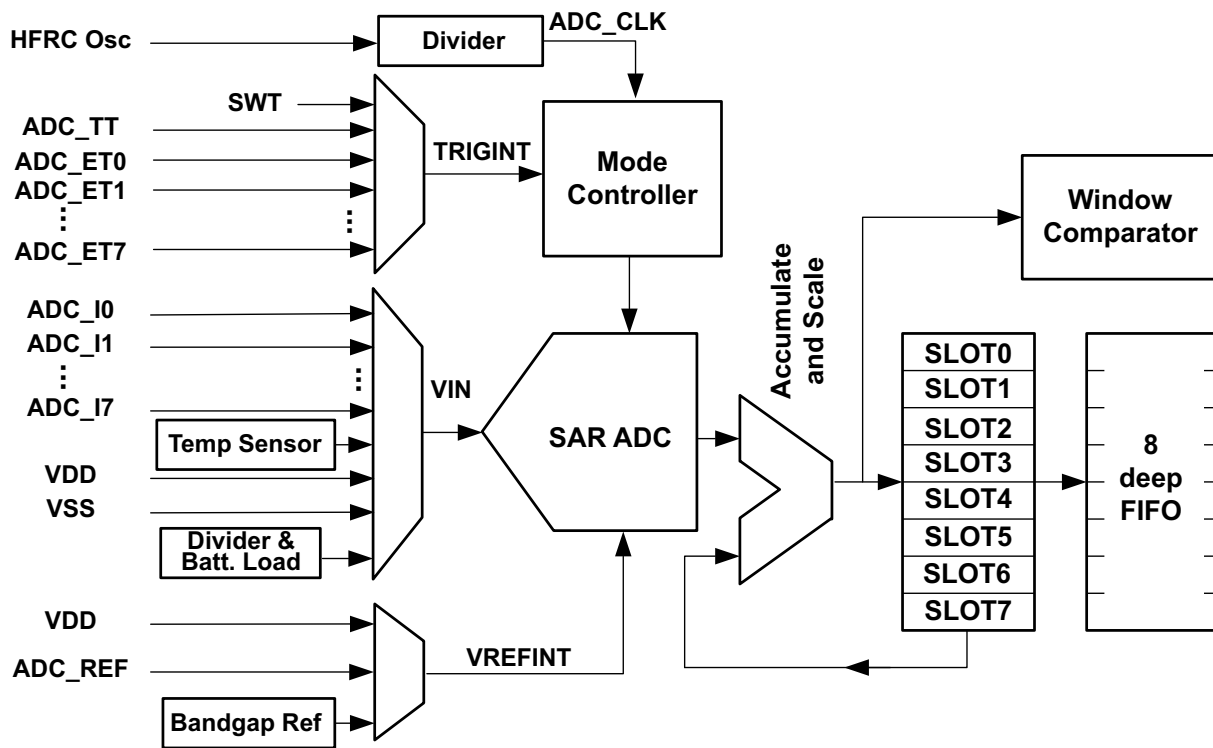


Figure 50. Block Diagram for ADC and Temperature Sensor

13.1 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 50.

Key features include:

- 13 user-selectable channels including:
 - external pins
 - internal voltages
 - voltage divider
 - 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

13.2 Functional Overview

The Apollo MCU 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.

13.2.1 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 24MHz HFRC:

- 12MHz
- 6MHz
- 3MHz
- 1.5MHz

13.2.2 13 Channel Analog Mux

As shown in Figure 50, the ADC block contains a 13 channel analog multiplexer on the input port to the analog to digital converter. Eight of the GPIO pins on the Apollo MCU 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:

- ADC_EXT0 external GPIO pin connection.
- ADC_EXT1 external GPIO pin connection.
- ADC_EXT2 external GPIO pin connection.
- ADC_EXT3 external GPIO pin connection.
- ADC_EXT4 external GPIO pin connection.
- ADC_EXT5 external GPIO pin connection.
- ADC_EXT6 external GPIO pin connection.
- ADC_EXT7 external GPIO pin connection.
- ADC_TEMP internal temperature sensor.
- ADC_VSS internal ground connection.
- ADC_VDD internal connection to input power rails.
- ADC_DIV3 internal voltage divide by 3 connection to the input power rail.
- ADC_DIV3_LOAD internal voltage divide by 3 connection to 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 “Voltage Divider and Switchable Battery Load” on page 319.

13.2.3 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. Seven of the GPIO pins on the Apollo MCU 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:

1. ADC_TRIG0 external GPIO pin connection.
2. ADC_TRIG1 external GPIO pin connection.

- Finally it is important to note that a trigger event applies to all enabled slots as a whole. Individual slots can not be separately triggered.

The window comparator enable will be discussed in a subsequent section, below. See “Window Comparator” on page 314. The number of samples to accumulate will also be explained in a subsequent section. See “Automatic Sample Accumulation and Scaling” on page 312.

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
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 “13 Channel Analog Mux” on page 310.

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

13.2.6 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 sample were accumulated to produce the results. Table 490 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 490: 10.6 ADC Sample Format

1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
10-bit Integer										6-bit Fraction					

Each slot contains a 17-bit accumulator as shown in Table 491, “Per Slot Sample Accumulator,” on page 312. 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:

$$128 \times 1023 = 128 \times 1023 = 130944 = 2^{17} - 128, \text{ hence the 17 bit accumulator.}$$

Table 491: Per Slot Sample Accumulator

1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
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 492 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 492: Accumulator Scaling

# Samples	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
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									

13.2.7 Eight Entry Result FIFO

All results written to the FIFO have exactly the same format as shown in Table 493. 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 493: FIFO Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
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 conver-

- sion 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.

13.2.8 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. To put this another way, 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 494: Window Comparator Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
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 be 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.

13.3 Operating Modes and the Mode Controller

The mode controller of Figure 50 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 from 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 configuration register. Setting this bit to zero has many of the effects of a software reset, such as resetting 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 50 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 51

An ADC conversion is the process of averaging measurements following one or more scans for each slot that is enabled.

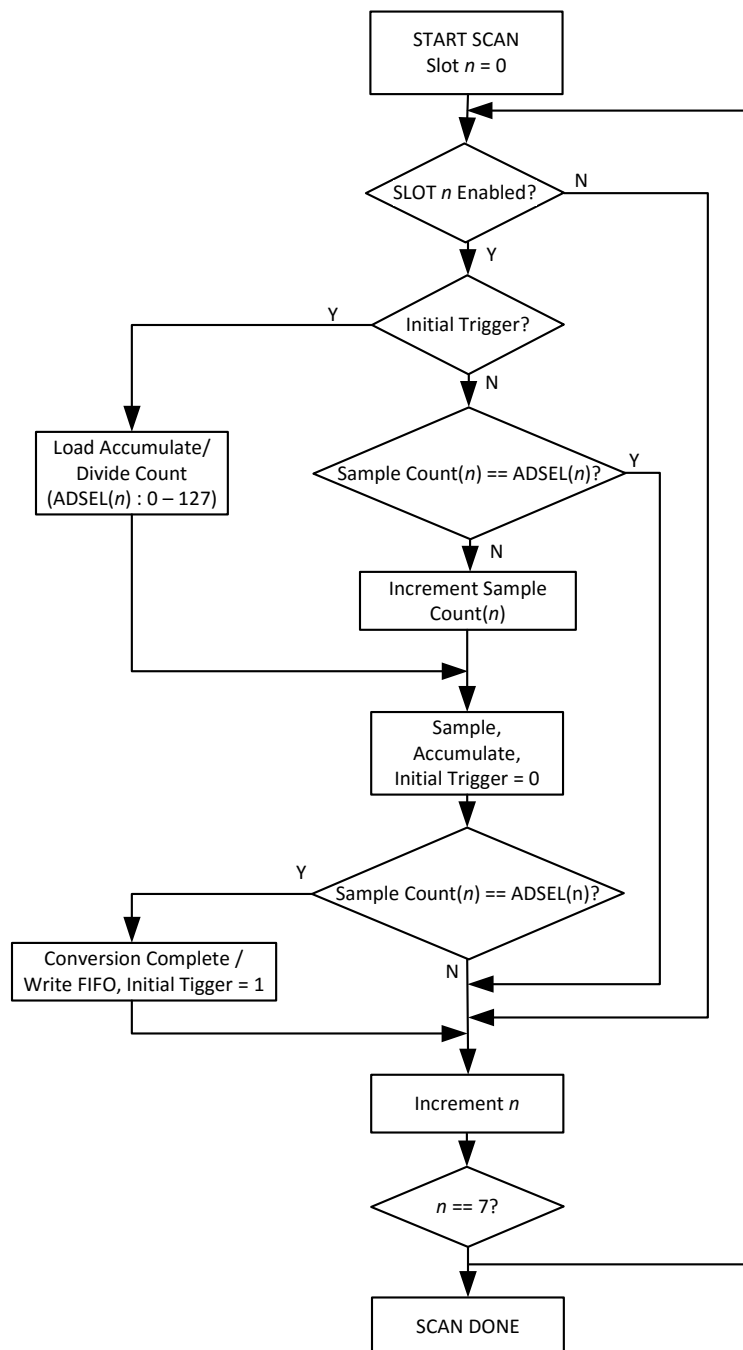


Figure 51. Scan Flowchart

13.3.1 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 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.

13.3.2 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.

13.3.3 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.

13.4 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 Section 13.2.8

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.

13.5 Mode Controller State Machine

The ADC architecture eliminates CPU and software overhead for managing the power switches, clock enables, data collection, averaging and data threshold comparisons such that the system including the CPU and all other peripheral circuitry may be powered off while a configured number of data samples is being periodically collected and averaged or until the averaged data value has crossed a configured threshold. This reduces the total energy consumed supporting the ADC functionality to only the energy consumed in the ADC circuitry itself and its supporting control logic during one of the periodic sample data conversion events and reduces the power in the system effectively to state retained leakage power and the trigger wake-up circuitry in the time between data sample conversion events.

Data collection and averaging can occur autonomously without requiring CPU interaction for multiple independent data measurement slots having, each processing a configurable number of samples on one of the analog data channels before storing the processed data in the FIFO. Upon completion of the data sample collection and averaging, the resultant data is stored in the software accessible FIFO along with its associated measurement slot number. With the slot number, valid entry queue count and associated measurement data stored in FIFO, software is simplified such that it does not need to maintain the remaining sample counts per channel or perform any filtering or averaging while the data is being collected thereby further reducing overall energy consumption.

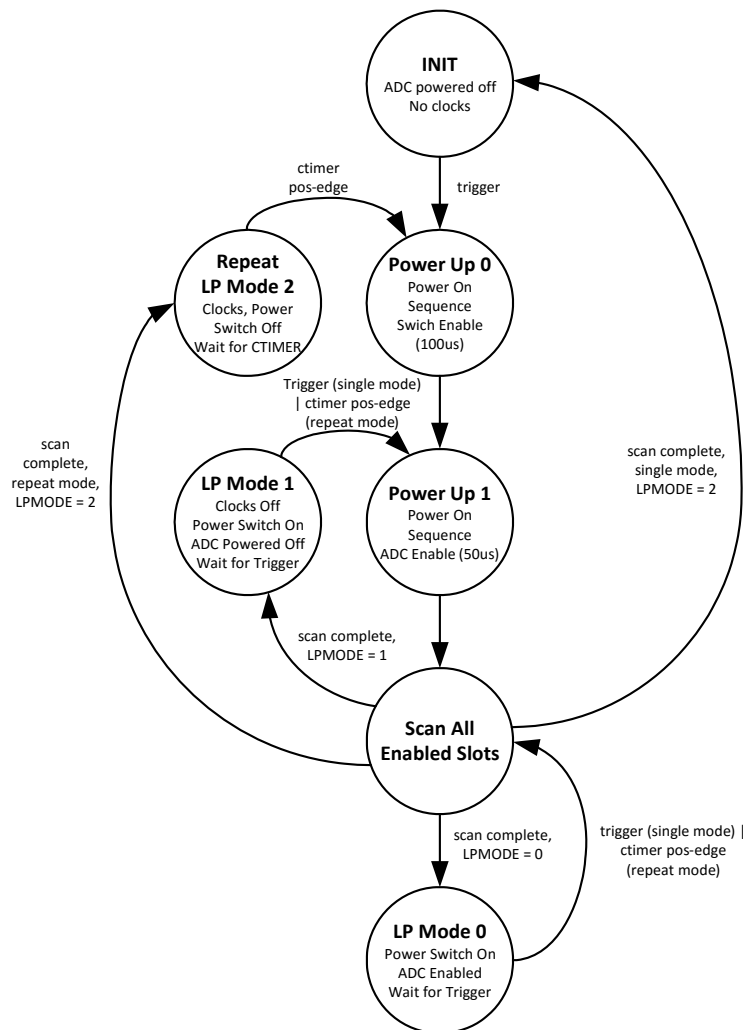
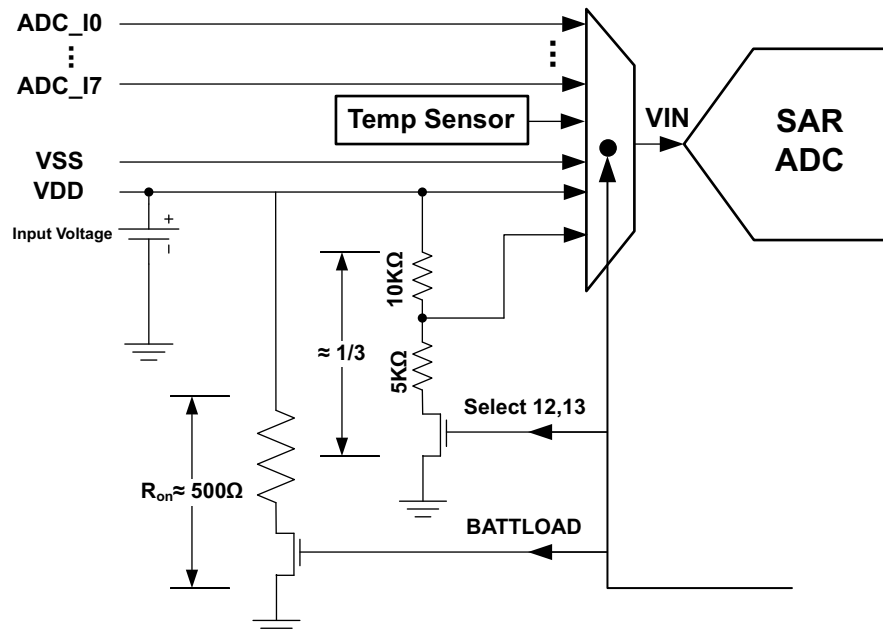


Figure 52. ADC State Diagram

13.6 Voltage Divider and Switchable Battery Load

The Apollo MCU'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 MCU chip. The voltage divider is only switched on when one of the active slots is selecting analog mux channel 12 or 13. 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.



The switchable load resistor is enabled by the BATTLOAD bit in the CFG register for the ADC. This feature is used to help estimate the health of the battery chemistry by estimating the internal resistance of the battery.

13.7 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.

13.7.1 Register Memory Map

Table 495: 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

13.7.2 ADC Registers

13.7.2.1 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 496: CFG Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD					CLKSEL			RSVD			TRIGPOL	TRIGSEL				RSVD						REFSEL		OPMODE			LPMODE		RPTEN	TMPSPWR	ADCEN

Table 497: 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. OFF = 0x0 - Low Power Mode. 12MHZ = 0x1 - 12 MHz ADC clock. 6MHZ = 0x2 - 6 MHz ADC clock. 3MHZ = 0x3 - 3 MHz ADC clock. 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. RISING_EDGE = 0x0 - Trigger on rising edge. FALLING_EDGE = 0x1 - Trigger on falling edge.
19:16	TRIGSEL	0x0	RW	Select the ADC trigger source. EXT0 = 0x0 - Off chip External Trigger0 (ADC_ET0) EXT1 = 0x1 - Off chip External Trigger1 (ADC_ET1) EXT2 = 0x2 - Off chip External Trigger2 (ADC_ET2) EXT3 = 0x3 - Off chip External Trigger3 (ADC_ET3) EXT4 = 0x4 - Off chip External Trigger4 (ADC_ET4) EXT5 = 0x5 - Off chip External Trigger5 (ADC_ET5) EXT6 = 0x6 - Off chip External Trigger6 (ADC_ET6) EXT7 = 0x7 - Off chip External Trigger7 (ADC_ET7) SWT = 0x8 - Software Trigger

Table 497: CFG Register Bits

Bit	Name	Reset	RW	Description
15:10	RSVD	0x0	RO	RESERVED.
9:8	REFSEL	0x0	RW	Select the ADC reference voltage. INTERNAL = 0x0 - Internal Bandgap Reference Voltage VDD = 0x1 - Select VDD as the ADEC reference voltage. ADCREF = 0x2 - Off Chip Reference (ADC_REF)
7:5	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 800 kSps depending on the trigger or repeating mode rate. A 1.5 MHz ADC clock can result in a sample rate up to 125 kSps. NOTE: All other values not specified below are undefined. SAMPLE_RATE_LE_125KSPS = 0x0 - Sample Rate $\leq$ 125 kSps SAMPLE_RATE_125K_800KSPS = 0x2 - Sample Rate 125 kSps to 800 kSps. NOTE: When running on a 12 MHz ADC clock, OPMODE settings other than 0x0 are only valid when Vdd is $\geq$ 2.0 V.
4:3	LPMODE	0x0	RW	Select power mode to enter between active scans. 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. 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). 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). MODE_UNDEFINED = 0x3 - Undefined Mode (2'b11)
2	RPTEN	0x0	RW X	This bit enables Repeating Scan Mode. SINGLE_SCAN = 0x0 - In Single Scan Mode, the ADC will complete a single scan upon each trigger event. 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). DIS = 0x0 - Power down the temperature sensor. 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. DIS = 0x0 - Disable the ADC module. EN = 0x1 - Enable the ADC module.

13.7.2.2 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 498: STAT Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																													PWDSTAT		

Table 499: 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. ON = 0x0 - Powered on. SWITCH_ON_SAR_OFF = 0x1 - Power switch on, ADC Low Power Mode 1. POWER_SWITCH_OFF = 0x2 - Power switch off, ADC disabled.

13.7.2.3 SWT Register

Software trigger

OFFSET: 0x00000008**INSTANCE 0 ADDRESS: 0x50008008**

This register enables initiating an ADC scan through software.

Table 500: SWT Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	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			
RSVD																					SWT									

Table 501: 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. GEN_SW_TRIGGER = 0x37 - Writing this value generates a software trigger.

13.7.2.4 SL0CFG Register

Slot 0 Configuration Register

OFFSET: 0x0000000C

INSTANCE 0 ADDRESS: 0x5000800C

Slot 0 Configuration Register

Table 502: SL0CFG Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD				ADSELO		RSVD				THSELO		RSVD				CHSELO				RSVD				WCEN0	SLEN0						

Table 503: SL0CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.
26:24	ADSELO	0x0	RW	Select the number of measurements to average in the accumulate divide module for this slot. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot.

Table 503: SL0CFG Register Bits

Bit	Name	Reset	RW	Description
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. 1_ADC_CLK = 0x0 - 1 ADC clock cycle. 2_ADC_CLKS = 0x1 - 2 ADC clock cycles. 4_ADC_CLKS = 0x2 - 4 ADC clock cycles. 8_ADC_CLKS = 0x3 - 8 ADC clock cycles. 16_ADC_CLKS = 0x4 - 16 ADC clock cycles. 32_ADC_CLKS = 0x5 - 32 ADC clock cycles. 64_ADC_CLKS = 0x6 - 64 ADC clock cycles. 128_ADC_CLKS = 0x7 - 128 ADC clock cycles.
15:12	RSVD	0x0	RO	RESERVED.
11:8	CHSEL0	0x0	RW	Select one of the 13 channel inputs for this slot. EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection. EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection. EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection. EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection. EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection. EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection. EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection. EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection. TEMP = 0x8 - ADC_TEMP internal temperature sensor. VDD = 0x9 - ADC_VDD internal power rail connection. VSS = 0xA - ADC_VSS internal ground connection. 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. WCEN = 0x1 - Enable the window compare for slot 0.
0	SLEN0	0x0	RW	This bit enables slot 0 for ADC conversions. SLEN = 0x1 - Enable slot 0 for ADC conversions.

13.7.2.5 SL1CFG Register

Slot 1 Configuration Register

OFFSET: 0x00000010

INSTANCE 0 ADDRESS: 0x50008010

Slot 1 Configuration Register

Table 504: SL1CFG Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	0	0	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
RSVD				ADSEL1				RSVD				THSEL1				RSVD				CHSEL1				RSVD				WCEN1		SLEN1	

Table 505: SL1CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.
26:24	ADSEL1	0x0	RW	Select the number of measurements to average in the accumulate divide module for this slot. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot.
23:19	RSVD	0x0	RO	RESERVED.

Table 505: SL1CFG Register Bits

Bit	Name	Reset	RW	Description
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. 1_ADC_CLK = 0x0 - 1 ADC clock cycle. 2_ADC_CLKS = 0x1 - 2 ADC clock cycles. 4_ADC_CLKS = 0x2 - 4 ADC clock cycles. 8_ADC_CLKS = 0x3 - 8 ADC clock cycles. 16_ADC_CLKS = 0x4 - 16 ADC clock cycles. 32_ADC_CLKS = 0x5 - 32 ADC clock cycles. 64_ADC_CLKS = 0x6 - 64 ADC clock cycles. 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. EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection. EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection. EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection. EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection. EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection. EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection. EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection. EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection. TEMP = 0x8 - ADC_TEMP internal temperature sensor. VDD = 0x9 - ADC_VDD internal power rail connection. VSS = 0xA - ADC_VSS internal ground connection. 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. WCEN = 0x1 - Enable the window compare for slot 1.
0	SLEN1	0x0	RW	This bit enables slot 1 for ADC conversions. SLEN = 0x1 - Enable slot 1 for ADC conversions.

Table 507: 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. 1_ADC_CLK = 0x0 - 1 ADC clock cycle. 2_ADC_CLKS = 0x1 - 2 ADC clock cycles. 4_ADC_CLKS = 0x2 - 4 ADC clock cycles. 8_ADC_CLKS = 0x3 - 8 ADC clock cycles. 16_ADC_CLKS = 0x4 - 16 ADC clock cycles. 32_ADC_CLKS = 0x5 - 32 ADC clock cycles. 64_ADC_CLKS = 0x6 - 64 ADC clock cycles. 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. EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection. EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection. EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection. EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection. EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection. EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection. EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection. EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection. TEMP = 0x8 - ADC_TEMP internal temperature sensor. VDD = 0x9 - ADC_VDD internal power rail connection. VSS = 0xA - ADC_VSS internal ground connection. 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. WCEN = 0x1 - Enable the window compare for slot 2.
0	SLEN2	0x0	RW	This bit enables slot 2 for ADC conversions. SLEN = 0x1 - Enable slot 2 for ADC conversions.

Table 509: SL3CFG Register Bits

Bit	Name	Reset	RW	Description
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. 1_ADC_CLK = 0x0 - 1 ADC clock cycle. 2_ADC_CLKS = 0x1 - 2 ADC clock cycles. 4_ADC_CLKS = 0x2 - 4 ADC clock cycles. 8_ADC_CLKS = 0x3 - 8 ADC clock cycles. 16_ADC_CLKS = 0x4 - 16 ADC clock cycles. 32_ADC_CLKS = 0x5 - 32 ADC clock cycles. 64_ADC_CLKS = 0x6 - 64 ADC clock cycles. 128_ADC_CLKS = 0x7 - 128 ADC clock cycles.
15:12	RSVD	0x0	RO	RESERVED.
11:8	CHSEL3	0x0	RW	Select one of the 13 channel inputs for this slot. EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection. EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection. EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection. EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection. EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection. EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection. EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection. EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection. TEMP = 0x8 - ADC_TEMP internal temperature sensor. VDD = 0x9 - ADC_VDD internal power rail connection. VSS = 0xA - ADC_VSS internal ground connection. 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. WCEN = 0x1 - Enable the window compare for slot 3.
0	SLEN3	0x0	RW	This bit enables slot 3 for ADC conversions. SLEN = 0x1 - Enable slot 3 for ADC conversions.

13.7.2.8 SL4CFG Register

Slot 4 Configuration Register

OFFSET: 0x0000001C

INSTANCE 0 ADDRESS: 0x5000801C

Slot 4 Configuration Register

Table 510: SL4CFG Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	0	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
RSVD				ADSEL4				RSVD				THSEL4				RSVD				CHSEL4				RSVD				WCEN4	SLEN4		

Table 511: SL4CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.
26:24	ADSEL4	0x0	RW	Select the number of measurements to average in the accumulate divide module for this slot. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot.
23:19	RSVD	0x0	RO	RESERVED.

Table 511: SL4CFG Register Bits

Bit	Name	Reset	RW	Description
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. 1_ADC_CLK = 0x0 - 1 ADC clock cycle. 2_ADC_CLKS = 0x1 - 2 ADC clock cycles. 4_ADC_CLKS = 0x2 - 4 ADC clock cycles. 8_ADC_CLKS = 0x3 - 8 ADC clock cycles. 16_ADC_CLKS = 0x4 - 16 ADC clock cycles. 32_ADC_CLKS = 0x5 - 32 ADC clock cycles. 64_ADC_CLKS = 0x6 - 64 ADC clock cycles. 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. EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection. EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection. EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection. EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection. EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection. EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection. EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection. EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection. TEMP = 0x8 - ADC_TEMP internal temperature sensor. VDD = 0x9 - ADC_VDD internal power rail connection. VSS = 0xA - ADC_VSS internal ground connection. 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. WCEN = 0x1 - Enable the window compare for slot 4.
0	SLEN4	0x0	RW	This bit enables slot 4 for ADC conversions. SLEN = 0x1 - Enable slot 4 for ADC conversions.

13.7.2.9 SL5CFG Register

Slot 5 Configuration Register

OFFSET: 0x00000020

INSTANCE 0 ADDRESS: 0x50008020

Slot 5 Configuration Register

Table 512: SL5CFG Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD				ADSEL5		RSVD				THSEL5		RSVD			CHSEL5			RSVD				WCEN5	SLEN5								

Table 513: 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. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. AVG_128_MSRMTS = 0x7 - Average in 128 measurements in the accumulate divide module for this slot.
23:19	RSVD	0x0	RO	RESERVED.
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. 1_ADC_CLK = 0x0 - 1 ADC clock cycle. 2_ADC_CLKS = 0x1 - 2 ADC clock cycles. 4_ADC_CLKS = 0x2 - 4 ADC clock cycles. 8_ADC_CLKS = 0x3 - 8 ADC clock cycles. 16_ADC_CLKS = 0x4 - 16 ADC clock cycles. 32_ADC_CLKS = 0x5 - 32 ADC clock cycles. 64_ADC_CLKS = 0x6 - 64 ADC clock cycles. 128_ADC_CLKS = 0x7 - 128 ADC clock cycles.

Table 513: SL5CFG Register Bits

Bit	Name	Reset	RW	Description
15:12	RSVD	0x0	RO	RESERVED.
11:8	CHSEL5	0x0	RW	Select one of the 13 channel inputs for this slot. EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection. EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection. EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection. EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection. EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection. EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection. EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection. EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection. TEMP = 0x8 - ADC_TEMP internal temperature sensor. VDD = 0x9 - ADC_VDD internal power rail connection. VSS = 0xA - ADC_VSS internal ground connection. 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. WCEN = 0x1 - Enable the window compare for slot 5.
0	SLEN5	0x0	RW	This bit enables slot 5 for ADC conversions. SLEN = 0x1 - Enable slot 5 for ADC conversions.

13.7.2.10 SL6CFG Register

Slot 6 Configuration Register

OFFSET: 0x00000024

INSTANCE 0 ADDRESS: 0x50008024

Slot 6 Configuration Register

Table 514: SL6CFG Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	0	0	0	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
RSVD				ADSEL6				RSVD				THSEL6				RSVD				CHSEL6				RSVD				WCEN6		SLEN6	

Table 515: 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. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. 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. 1_ADC_CLK = 0x0 - 1 ADC clock cycle. 2_ADC_CLKS = 0x1 - 2 ADC clock cycles. 4_ADC_CLKS = 0x2 - 4 ADC clock cycles. 8_ADC_CLKS = 0x3 - 8 ADC clock cycles. 16_ADC_CLKS = 0x4 - 16 ADC clock cycles. 32_ADC_CLKS = 0x5 - 32 ADC clock cycles. 64_ADC_CLKS = 0x6 - 64 ADC clock cycles. 128_ADC_CLKS = 0x7 - 128 ADC clock cycles.

Table 515: SL6CFG Register Bits

Bit	Name	Reset	RW	Description
15:12	RSVD	0x0	RO	RESERVED.
11:8	CHSEL6	0x0	RW	Select one of the 13 channel inputs for this slot. EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection. EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection. EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection. EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection. EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection. EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection. EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection. EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection. TEMP = 0x8 - ADC_TEMP internal temperature sensor. VDD = 0x9 - ADC_VDD internal power rail connection. VSS = 0xA - ADC_VSS internal ground connection. 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. WCEN = 0x1 - Enable the window compare for slot 6.
0	SLEN6	0x0	RW	This bit enables slot 6 for ADC conversions. SLEN = 0x1 - Enable slot 6 for ADC conversions.

13.7.2.11 SL7CFG Register

Slot 7 Configuration Register

OFFSET: 0x00000028

INSTANCE 0 ADDRESS: 0x50008028

Slot 7 Configuration Register

Table 516: SL7CFG Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD				ADSEL7		RSVD				THSEL7		RSVD			CHSEL7			RSVD				WCEN7	SLEN7								

Table 517: SL7CFG Register Bits

Bit	Name	Reset	RW	Description
31:27	RSVD	0x0	RO	RESERVED.

Table 517: 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. AVG_1_MSRMT = 0x0 - Average in 1 measurement in the accumulate divide module for this slot. AVG_2_MSRMTS = 0x1 - Average in 2 measurements in the accumulate divide module for this slot. AVG_4_MSRMTS = 0x2 - Average in 4 measurements in the accumulate divide module for this slot. AVG_8_MSRMT = 0x3 - Average in 8 measurements in the accumulate divide module for this slot. AVG_16_MSRMTS = 0x4 - Average in 16 measurements in the accumulate divide module for this slot. AVG_32_MSRMTS = 0x5 - Average in 32 measurements in the accumulate divide module for this slot. AVG_64_MSRMTS = 0x6 - Average in 64 measurements in the accumulate divide module for this slot. 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. 1_ADC_CLK = 0x0 - 1 ADC clock cycle. 2_ADC_CLKS = 0x1 - 2 ADC clock cycles. 4_ADC_CLKS = 0x2 - 4 ADC clock cycles. 8_ADC_CLKS = 0x3 - 8 ADC clock cycles. 16_ADC_CLKS = 0x4 - 16 ADC clock cycles. 32_ADC_CLKS = 0x5 - 32 ADC clock cycles. 64_ADC_CLKS = 0x6 - 64 ADC clock cycles. 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. EXT0 = 0x0 - ADC_EXT0 external GPIO pin connection. EXT1 = 0x1 - ADC_EXT1 external GPIO pin connection. EXT2 = 0x2 - ADC_EXT2 external GPIO pin connection. EXT3 = 0x3 - ADC_EXT3 external GPIO pin connection. EXT4 = 0x4 - ADC_EXT4 external GPIO pin connection. EXT5 = 0x5 - ADC_EXT5 external GPIO pin connection. EXT6 = 0x6 - ADC_EXT6 external GPIO pin connection. EXT7 = 0x7 - ADC_EXT7 external GPIO pin connection. TEMP = 0x8 - ADC_TEMP internal temperature sensor. VDD = 0x9 - ADC_VDD internal power rail connection. VSS = 0xA - ADC_VSS internal ground connection. 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. WCEN = 0x1 - Enable the window compare for slot 7.

Table 517: SL7CFG Register Bits

Bit	Name	Reset	RW	Description
0	SLEN7	0x0	RW	This bit enables slot 7 for ADC conversions. SLEN = 0x1 - Enable slot 7 for ADC conversions.

13.7.2.12 WLIM Register

Window Comparator Limits Register

OFFSET: 0x0000002C**INSTANCE 0 ADDRESS: 0x5000802C**

Window Comparator Limits Register

Table 518: WLIM Register

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	0	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
ULIM																LLIM															

Table 519: WLIM Register Bits

Bit	Name	Reset	RW	Description
31:16	ULIM	0x0	RW	Sets the upper limit for the window comparator.
15:0	LLIM	0x0	RW	Sets the lower limit for the window comparator.

13.7.2.13 FIFO 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 520: FIFO Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD_27				SLOTNUM				RSVD_20				COUNT				DATA															

Table 521: 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.

13.7.2.14 INTEN 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 522: INTEN Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																										WCNC	WCXC	FIFOVR2	FIFOVR1	SCNCMP	CNVCMP

Table 523: INTEN Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED.
5	WCINC	0x0	RW	Window comparator voltage incursion interrupt. WCINCINT = 0x1 - Window comparator voltage incursion interrupt.
4	WCEXC	0x0	RW	Window comparator voltage excursion interrupt. WCEXCINT = 0x1 - Window comparator voltage excursion interrupt.
3	FIFOVR2	0x0	RW	FIFO 100% full interrupt. FIFOFULLINT = 0x1 - FIFO 100% full interrupt.

Bit	Name	Reset	RW	Description
2	FIFOVR1	0x0	RW	FIFO 75% full interrupt. FIFO75INT = 0x1 - FIFO 75% full interrupt.
1	SCNCMP	0x0	RW	ADC scan complete interrupt. SCNCMPINT = 0x1 - ADC scan complete interrupt.
0	CNVCMP	0x0	RW	ADC conversion complete interrupt. CNVCMPINT = 0x1 - ADC conversion complete interrupt.

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

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0	
RSVD																											WCINC	WCXC	FIFOVR2	FIFOVR1	SCNCMP	CNVCMP

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED.
5	WCINC	0x0	RW	Window comparator voltage incursion interrupt. WCINCINT = 0x1 - Window comparitor voltage incursion interrupt.
4	WCEXC	0x0	RW	Window comparator voltage excursion interrupt. WCEXCINT = 0x1 - Window comparitor voltage excursion interrupt.
3	FIFOVR2	0x0	RW	FIFO 100% full interrupt. FIFOFULLINT = 0x1 - FIFO 100% full interrupt.
2	FIFOVR1	0x0	RW	FIFO 75% full interrupt. FIFO75INT = 0x1 - FIFO 75% full interrupt.
1	SCNCMP	0x0	RW	ADC scan complete interrupt. SCNCMPINT = 0x1 - ADC scan complete interrupt.

Table 525: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
0	CNVCMP	0x0	RW	ADC conversion complete interrupt. CNVCMPINT = 0x1 - ADC conversion complete interrupt.

13.7.2.16 INTCLR 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 526: INTCLR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																										WCINC	WCEXC	FIFOVR2	FIFOVR1	SCNCMP	CNVCMP

Table 527: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED.
5	WCINC	0x0	RW	Window comparator voltage incursion interrupt. WCINCINT = 0x1 - Window comparator voltage incursion interrupt.
4	WCEXC	0x0	RW	Window comparator voltage excursion interrupt. WCEXCINT = 0x1 - Window comparator voltage excursion interrupt.
3	FIFOVR2	0x0	RW	FIFO 100% full interrupt. FIFOFULLINT = 0x1 - FIFO 100% full interrupt.
2	FIFOVR1	0x0	RW	FIFO 75% full interrupt. FIFO75INT = 0x1 - FIFO 75% full interrupt.
1	SCNCMP	0x0	RW	ADC scan complete interrupt. SCNCMPINT = 0x1 - ADC scan complete interrupt.
0	CNVCMP	0x0	RW	ADC conversion complete interrupt. CNVCMPINT = 0x1 - ADC conversion complete interrupt.

13.7.2.17 INTSET 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 528: INTSET Register

3	3	2	2	2	2	2	2	2	2	2	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
RSVD																							WCINC	WCEXC	FIFOVR2	FIFOVR1	SCNCMP	CNVCMP			

Table 529: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:6	RSVD	0x0	RO	RESERVED.
5	WCINC	0x0	RW	Window comparator voltage incursion interrupt. WCINCINT = 0x1 - Window comparator voltage incursion interrupt.
4	WCEXC	0x0	RW	Window comparator voltage excursion interrupt. WCEXCINT = 0x1 - Window comparator voltage excursion interrupt.
3	FIFOVR2	0x0	RW	FIFO 100% full interrupt. FIFOFULLINT = 0x1 - FIFO 100% full interrupt.
2	FIFOVR1	0x0	RW	FIFO 75% full interrupt. FIFO75INT = 0x1 - FIFO 75% full interrupt.
1	SCNCMP	0x0	RW	ADC scan complete interrupt. SCNCMPINT = 0x1 - ADC scan complete interrupt.
0	CNVCMP	0x0	RW	ADC conversion complete interrupt. CNVCMPINT = 0x1 - ADC conversion complete interrupt.

14. Voltage Comparator Module

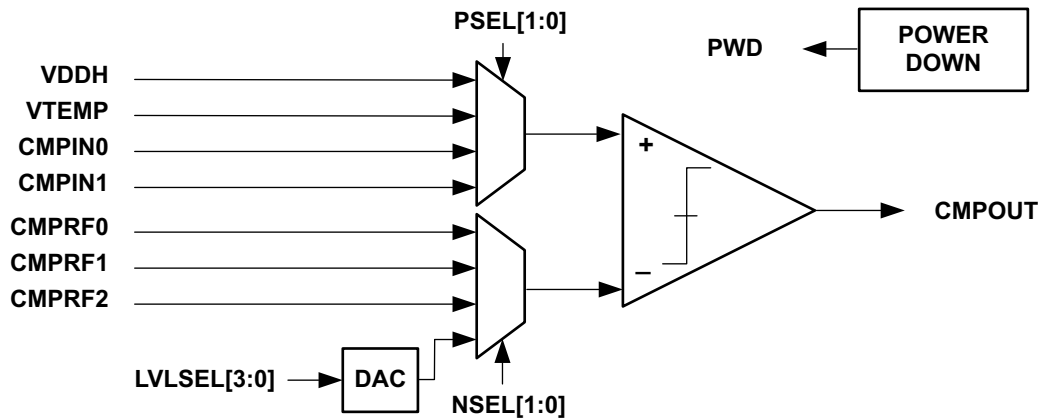


Figure 53. Block diagram for the Voltage Comparator Module

14.1 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 (VDD), 2) the PTAT voltage from the temperature sensor (VTEMP), or 3) two external voltage channels (CMPIN0 and 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, CMPRF2), or 2) the internally generated reference (VREFINT). 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 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. 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.

14.2 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.

14.2.1 Register Memory Map

Table 530: 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

14.2.2 VCOMP Registers

14.2.2.1 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 531: CFG Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD												LVLSEL				RSVD				NSEL	RSVD				PSEL						

Table 532: 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. 0P58V = 0x0 - Set Reference input to 0.58 Volts. 0P77V = 0x1 - Set Reference input to 0.77 Volts. 0P97V = 0x2 - Set Reference input to 0.97 Volts. 1P16V = 0x3 - Set Reference input to 1.16 Volts. 1P35V = 0x4 - Set Reference input to 1.35 Volts. 1P55V = 0x5 - Set Reference input to 1.55 Volts. 1P74V = 0x6 - Set Reference input to 1.74 Volts. 1P93V = 0x7 - Set Reference input to 1.93 Volts. 2P13V = 0x8 - Set Reference input to 2.13 Volts. 2P32V = 0x9 - Set Reference input to 2.32 Volts. 2P51V = 0xA - Set Reference input to 2.51 Volts. 2P71V = 0xB - Set Reference input to 2.71 Volts. 2P90V = 0xC - Set Reference input to 2.90 Volts. 3P09V = 0xD - Set Reference input to 3.09 Volts. 3P29V = 0xE - Set Reference input to 3.29 Volts. 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. CMPRF0 = 0x0 - Use comparator reference 0 for reference input. CMPRF1 = 0x1 - Use comparator reference 1 for reference input. CMPRF2 = 0x3 - Use comparator reference 2 for reference input. DAC = 0x3 - Use DAC output selected by LVLSEL for reference input.
7:2	RSVD	0x0	RO	This bitfield is reserved for future use.
1:0	PSEL	0x0	RW	This bitfield selects the positive input to the comparator. VDDADJ = 0x0 - Use VDDADJ for the positive input. VTEMP = 0x1 - Use the temperature sensor output for the positive input. VEXT1 = 0x2 - Use external voltage 1 for positive input. VEXT2 = 0x3 - Use external voltage 1 for positive input.

Bit	Name	Reset	RW	Description
31:0	PWDKEY	0x0	RW	Key register value. Key = 0x37 - Key

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	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
RSVD																													OUTH	OUTLOW	

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bitfield is reserved for future use.
1	OUTH1	0x0	RW	This bit is the vcompout high interrupt.
0	OUTLOW	0x0	RW	This bit is the vcompout low interrupt.

14.2.2.5 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 539: INTSTAT Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																													OUTH	OUTLOW	

Table 540: INTSTAT Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bitfield is reserved for future use.
1	OUTH1	0x0	RW	This bit is the vcompout high interrupt.
0	OUTLOW	0x0	RW	This bit is the vcompout low interrupt.

14.2.2.6 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 541: INTCLR Register

3 1	3 0	2 9	2 8	2 7	2 6	2 5	2 4	2 3	2 2	2 1	2 0	1 9	1 8	1 7	1 6	1 5	1 4	1 3	1 2	1 1	1 0	0 9	0 8	0 7	0 6	0 5	0 4	0 3	0 2	0 1	0 0
RSVD																													OUTH	OUTLOW	

Table 542: INTCLR Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bitfield is reserved for future use.
1	OUTH1	0x0	RW	This bit is the vcompout high interrupt.
0	OUTLOW	0x0	RW	This bit is the vcompout low interrupt.

14.2.2.7 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 543: INTSET Register

3	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	0	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
RSVD																												OUTH	OUTLOW		

Table 544: INTSET Register Bits

Bit	Name	Reset	RW	Description
31:2	RSVD	0x0	RO	This bitfield is reserved for future use.
1	OUTH1	0x0	RW	This bit is the vcompout high interrupt.
0	OUTLOW	0x0	RW	This bit is the vcompout low interrupt.

15. Voltage Regulator Module

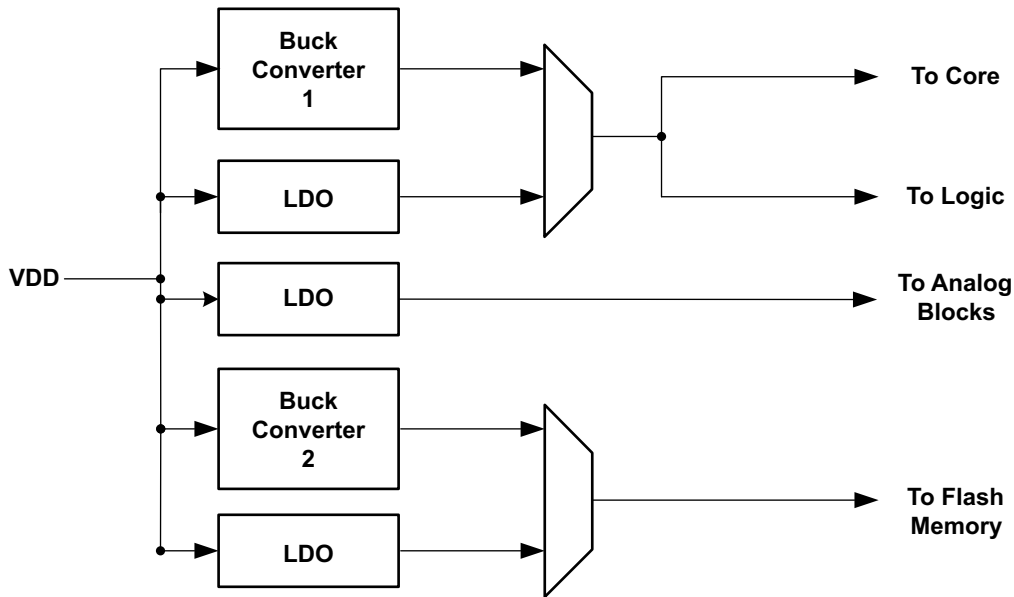


Figure 54. Block Diagram for the Voltage Regulator Module

15.1 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 >90%. 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 MCU. When the Apollo MCU 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.

16. Electrical Characteristics

For all tables $T_A = -40^{\circ}\text{C}$ to 85°C , Typical values at 25°C , unless otherwise specified.

16.1 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 545: Absolute Maximum Ratings

Symbol	Parameter	Test Conditions	Min	Max	Unit
V_{DDP}	Pad supply voltage		-	3.8	V
V_{DDH}	Digital supply voltage		-	3.8	V
V_{DDA}	Analog supply voltage		-	3.8	V
V_{IO}	Voltage on all input and output pins		0	V_{DDH}	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	-	TBD	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	-	TBD	mA
T_S	Storage temperature		-55	125	$^{\circ}\text{C}$
T_J	Junction temperature	Calculated assuming worst-case power consumption running Coremark, plus two high-side switches passing maximum static current.	-40	85	$^{\circ}\text{C}$
T_{OP}	Operating temperature		-40	85	$^{\circ}\text{C}$
T_{REFLOW}	Reflow temperature	Reflow Profile per JEDEC J-STD-020D	-	260	$^{\circ}\text{C}$
I_{LU}	Latch-up current		-	10	mA
V_{ESDHBM}	ESD Human Body Model (HBM)		-	2000	V
V_{ESDCDM}	ESD Charged Device Model (CDM)		-	500	V

⁽¹⁾ High side power switches are available on PAD3 and PAD4

⁽²⁾ A low side power switch is available on PAD11

16.2 Recommended Operating Conditions

Table 546: Recommended Operating Conditions^a

Symbol	Parameter	Min	Typ	Max	Unit
V _{DDP}	Pad supply voltage	1.8	-	3.8	V
V _{DDH}	Digital supply voltage	1.8	-	3.8	V
V _{DDA}	Analog supply voltage	1.8	-	3.8	V
T _A	Ambient operating temperature	-40	-	85	°C
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

a. V_{DD} = V_{DDP} = V_{DDA} = V_{DDH}

16.3 Current Consumption

Table 547: Current Consumption

Symbol	Parameter	Test Conditions ^{a,b}	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	TBD	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	293	nA
			1.8 V	270	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	343	nA
			1.8 V	320	nA

a. Core clock (HCLK) is 24 MHz for each parameter unless otherwise noted.

b. All values measured at 25°C

16.4 Power Mode Transitions

Table 548: 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

16.5 Clocks/Oscillators

Table 549: 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	25°C, STDEV of period	-	100	230	ps
T _{J_LFRC}	LFRC cycle-to-cycle jitter	25°C, STDEV of period	-	6	8	µs
DC _{HFRC}	HFRC duty cycle	HFRC/8 CLKOUT	40	50	60	%
DC _{LFRC}	LFRC duty cycle	25°C, LFRC/2	-	50	-	%
DC _{XT}	XT duty cycle	25°C	-	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

16.6 Analog-to-Digital Converter (ADC)

Table 550: 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
$I_{\text{ADC_RUN}}$	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
$I_{\text{ADC_RUN}}$	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
$T_{\text{ADCSAMPLE}}$	Sample time ^a		-	$15.5 / F_{\text{ADCLK}}$	$15 / F_{\text{ADCLK}}$	
$T_{\text{TRIG_TO_START0}}$	Delay from trigger to start of conversion – LPM0, Slot 0 HFRC running		-	$1 / F_{\text{ADCLK}}$	-	
$T_{\text{TRIG_TO_START2}}$	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 - Single Ended Input	-	0.05	0.12	%
SNR_{ADC}	Signal-to-noise ratio (SNR), Single-ended input	LDO mode - Single Ended Input	50	54	-	dB
$\text{SINAD}_{\text{ADC}}$	Signal-to-noise and distortion ration (SINAD), Single-ended input	LDO mode - Single Ended Input	50	54	-	dB
SFDR_{ADC}	Spurious-free dynamic range (SFDR), Single-ended input	LDO mode - Single Ended Input	62	67	-	dB

Table 550: Analog to Digital Converter (ADC)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
THD _{ADC}	Total harmonic distortion (THD), Single-ended input	Buck mode - Single Ended Input	-	0.06	0.12	%
SNR _{ADC}	Signal-to-noise ratio (SNR), Single-ended input	Buck mode - Single Ended Input	47	52	-	dB
SINAD _{ADC}	Signal-to-noise and distortion ratio (SINAD), Single-ended input	Buck mode - Single Ended Input	47	52	-	dB
SFDR _{ADC}	Spurious-free dynamic range (SFDR), Single-ended input	Buck mode - 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, Channel 4	-	1.8	-	LSB
DNL _{ADC}	Differential nonlinearity Full input range	Full input range, Channel 4	-	1	-	LSB
E _{ADC_OFFSET}	Offset error	Full input range, Channel 4	-	1	-	LSB
E _{ADC_GAIN}	Gain error	Full input range, Channel 4	-	0.3	-	LSB
E _{ADC_TOTAL}	Total unadjusted error Full input range	Full input range, Channel 4	-	1.3	-	LSB
Internal Temperature Sensor						
E _{TEMP}	Temperature sensor accuracy		-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		-	TBD	-	bits
T _{REFSETTLE}	Settling time		-	-	80	ns
I _{ADCREFIN}	Internal reference current	When reference is enabled for sampling	-	25	-	μA
External Reference						
V _{ADCREFEXT}	Voltage range		TBD	1.5	TBD	V
I _{ADCREFIN}	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

a. Max sample time is for less than or equal to eight samples. Typical sample rate is the average over greater than eight samples.

16.7 Buck Converter

Table 551: 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

16.8 Power-On RESET (POR) and Brown-Out Detector (BOD)

Table 552: Power-On Reset (POR) and Brown-Out Detector (BOD)

Symbol	Parameter	Min	Typ	Max	Unit
T_{VDDFS}	Falling slew rate on V_{DDP} / V_{DDA} / V_{DDH}	-	500	700	$\mu s/V$
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

16.9 Resets

Table 553: Resets

Symbol	Parameter	Min	Typ	Max	Unit
T _{RSTMIN}	Minimum nRST pulse width	-	1.3	-	μs
T _{POR}	POR to nRST delay	170	180	230	ms
T _{POR}	nRST deassertion to HRST deassertion	2.3	-	2.5	ms
T _{HWRST}	Delay from nRST assertion to HRST assertion	420	-	480	μs
T _{HWRST}	Delay from nRST deassertion to HRST deassertion	410	-	520	μs
T _{RST}	nRST pin reset delay from internal POR or BOD	-	30	-	μs
T _{SOFT}	Software initiated reset delay	320	TBD	660	μs

16.10 Voltage Comparator (VCOMP)

Table 554: 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

16.11 Internal DAC Reference for VCOMP

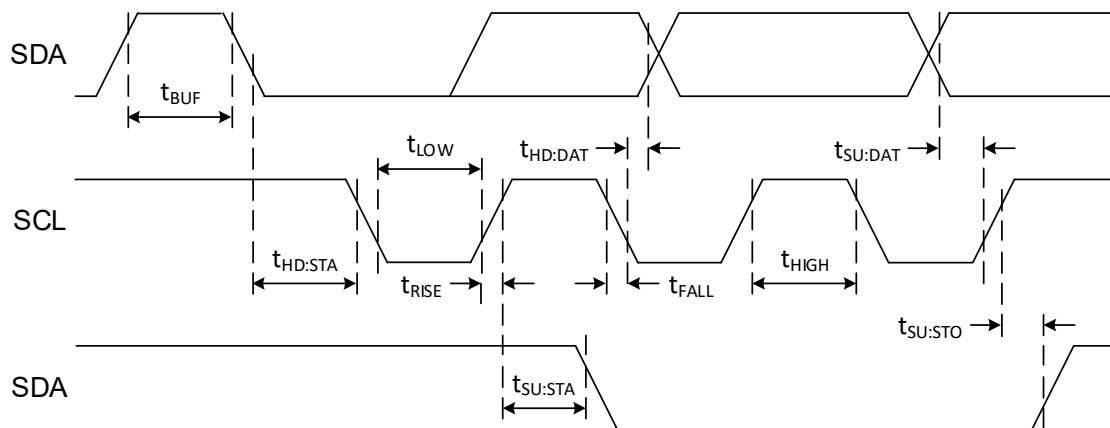
Table 555: Internal DAC Reference for VCOMP

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
N _{DACRES}	Resolution		-	4	-	bits

16.12 Inter-Integrated Circuit (I²C) Interface

Table 556: Inter-Integrated Circuit (I²C) Interface

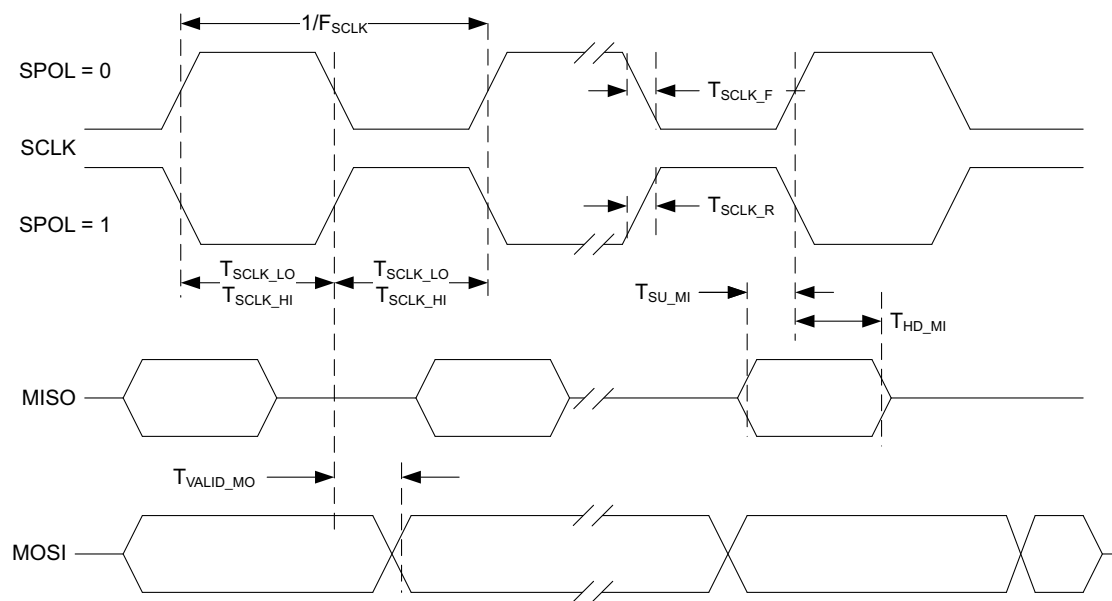
Symbol	Parameter	VCC	Min	Typ	Max	Unit
f _{SCL}	SCL input clock frequency	1.7 V - 3.6 V	10	-	1000	kHz
t _{LOW}	Low period of SCL clock	1.7 V - 3.6 V	1.3	-	-	μs
t _{HIGH}	High period of SCL clock	1.7 V - 3.6 V	600	-	-	ns
t _{RISE}	Rise time of SDA and SCL	1.7 V - 3.6 V		-	300	ns
t _{FALL}	Fall time of SDA and SCL	1.7 V - 3.6 V		-	300	ns
t _{HD:STA}	START condition hold time	1.7 V - 3.6 V	600	-	-	ns
t _{SU:STA}	START condition setup time	1.7 V - 3.6 V	600	-	-	ns
t _{SU:DAT}	SDA setup time	1.7 V - 3.6 V	100	-	-	ns
t _{HD:DAT}	SDA hold time	1.7 V - 3.6 V	0	-	-	ns
t _{SU:STO}	STOP condition setup time	1.7 V - 3.6 V	600	-	-	ns
t _{BUF}	Bus free time before a new transmission	1.7 V - 3.6 V	1.3	-	-	μs


Figure 55. I²C Timing

16.13 Serial Peripheral Interface (SPI) Master Interface

Table 557: Serial Peripheral Interface (SPI) Master Interface

Symbol	Parameter	Min	Typ	Max	Unit
F_{SCLK}	SCLK frequency range	TBD	8	TBD	MHz
B_{FIFO}	FIFO size	64	64	64	Bytes
T_{SCLK_LO}	Clock low time	TBD	-	-	TBD
T_{SCLK_HI}	Clock high time	TBD	-	-	TBD
T_{SCLK_R}	Clock rise time	TBD	-	-	ns
T_{SCLK_F}	Clock fall time	TBD	-	-	ns
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 56. SPI Master Mode, Phase = 0

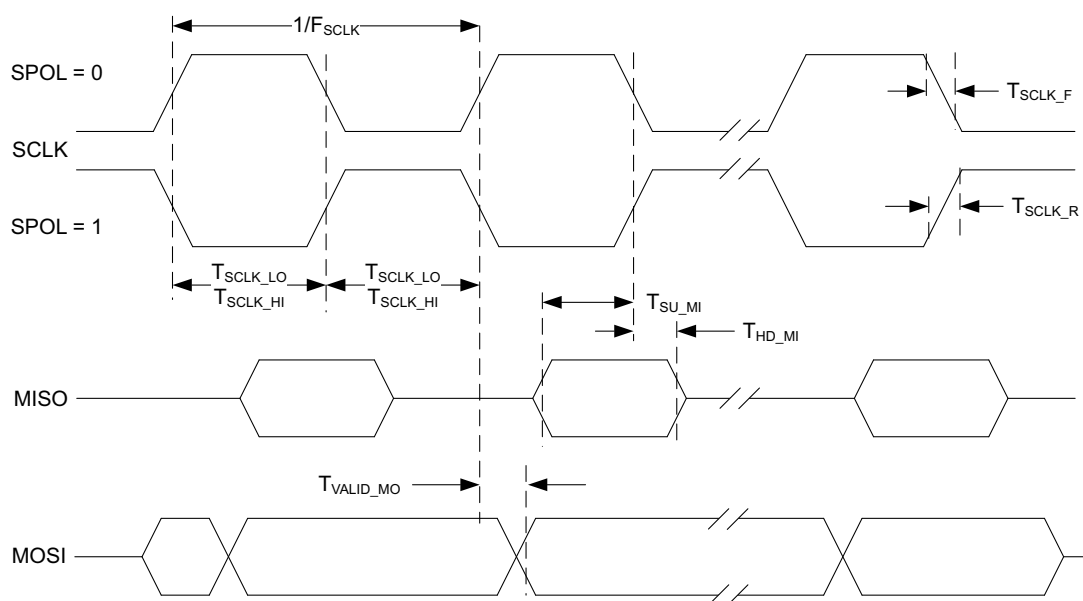
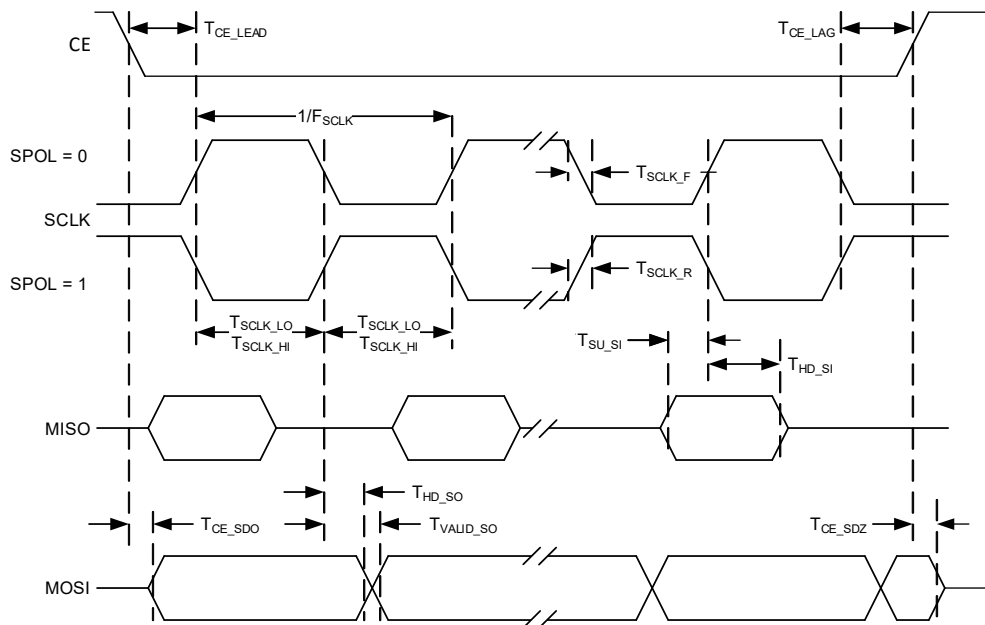


Figure 57. SPI Master Mode, Phase = 1

16.14 Serial Peripheral Interface (SPI) Slave Interface

Table 558: 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		TBD	-	-	TBD
T_{SCLK_HI}	Clock high time		TBD	-	-	TBD
T_{SCLK_R}	Clock rise time		TBD	-	-	ns
T_{SCLK_F}	Clock fall time		TBD	-	-	ns
T_{CE_LEAD}	Chip enable low to first SCLK edge		-	-	-	-
T_{CE_LAG}	Chip enable high to last SCLK edge		-	-	-	-
T_{CE_SDO}	Chip enable low to MISO data output		-	-	-	-
T_{CE_SDZ}	Chip enable high to MISO data tri-state		-	-	-	-
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 58. SPI Slave Mode, Phase = 0

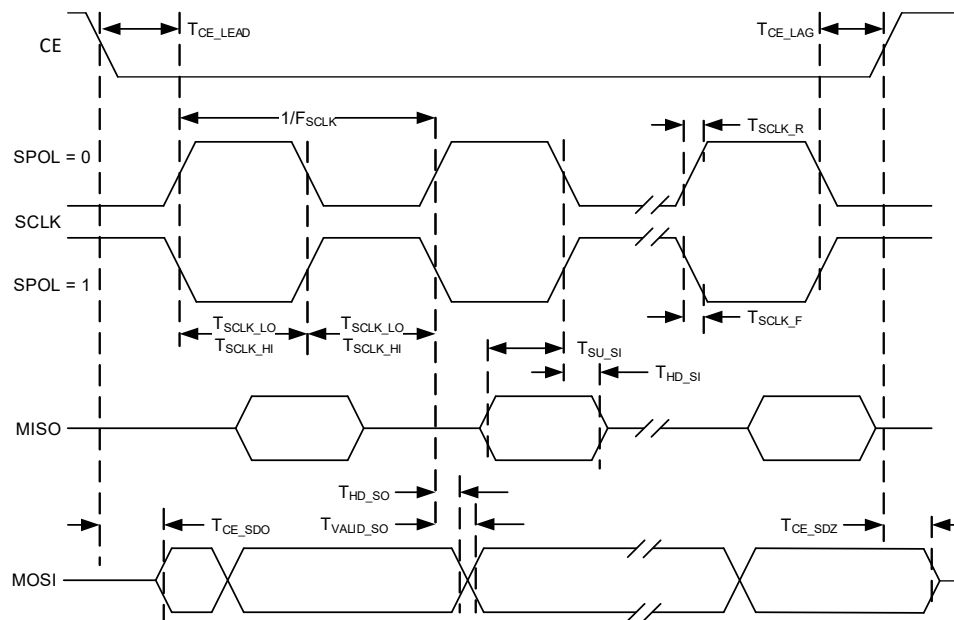


Figure 59. SPI Slave Mode, Phase = 1

16.15 Universal Asynchronous Receiver/Transmitter (UART)

Table 559: Universal Asynchronous Receiver/Transmitter (UART)

Symbol	Parameter	Min	Typ	Max	Unit
F_{BAUD}	UART baud rate	-	-	921600	bps

16.16 Counter/Timer (CTIMER)

Table 560: Counter/Timer (CTIMER)

Symbol	Parameter	Min	Typ	Max	Unit
F_{CTIMER}	Input frequency	-	-	24	MHz
T_{CTIMER}	Capture pulse width	100	-	-	ns

16.17 Flash Memory

Table 561: 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

16.18 General Purpose Input/Output (GPIO)

All GPIOs have Schmitt trigger inputs.

Table 562: 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
		5.1 mA 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 ² C pad pull-up resistance, RSEL = 0x00		-	1.5	-	kΩ
R _{PUI2C01}	I ² C pad pull-up resistance, RSEL = 0x01		-	6	-	kΩ
R _{PUI2C10}	I ² C pad pull-up resistance, RSEL = 0x10		-	12	-	kΩ
R _{PUI2C11}	I ² C pad pull-up resistance, RSEL = 0x11		-	24	-	kΩ
I _{IN}	Input pin leakage current		-	1	20	nA

Table 562: General Purpose Input/Output (GPIO)

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
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

16.19 Serial Wire Debug (SWD)

Table 563: Serial Wire Debug (SWD)

Symbol	Parameter	Min	Typ	Max	Unit
$T_{\text{SWDCK_HI}}$	SWDCK clock high period	0.085	-	500	μs
$T_{\text{SWDCK_LO}}$	SWDCK clock low period	0.085	-	500	μs
$T_{\text{OS_SWD}}$	SWDIO output skew to falling edge of SWDCLK	-5	-	45	ns
$T_{\text{SU_SWD}}$	Input setup time between SWDIO and rising edge SWDCK	65	-	-	ns
$T_{\text{HD_SWD}}$	Input hold time between SWDIO and rising edge SWDCK	125	-	-	ns

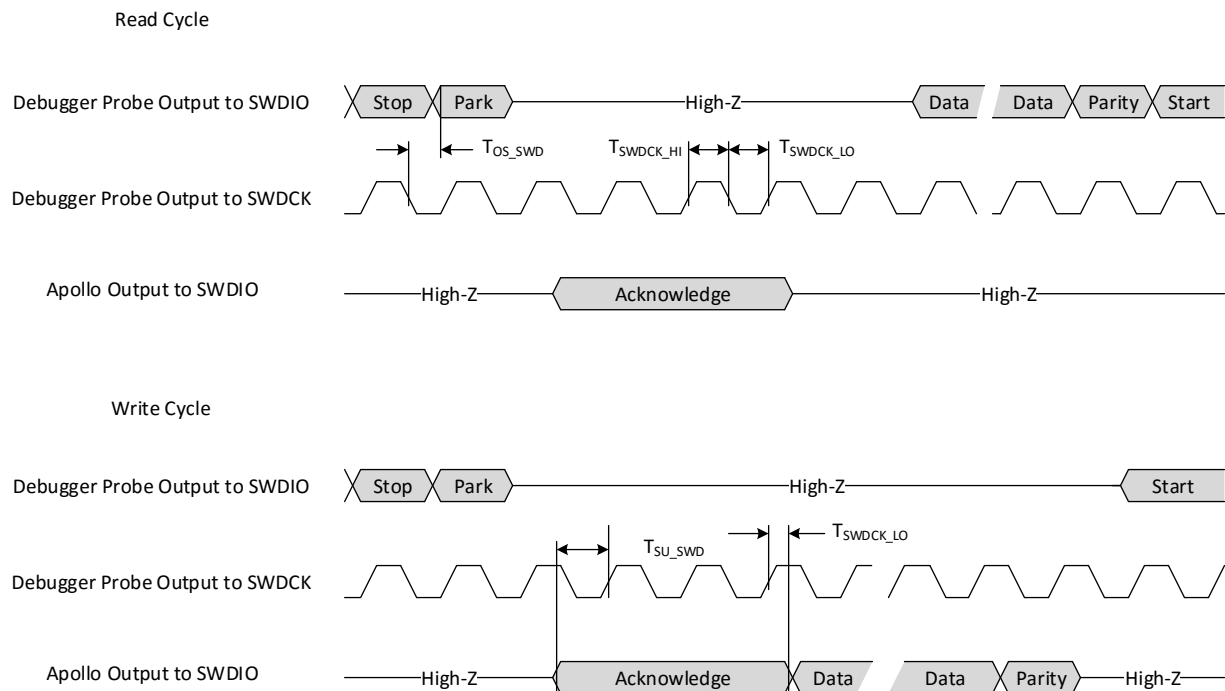


Figure 60. Serial Wire Debug Timing

16.20 Serial Wire Output (SWO)

Table 564: Serial Wire Output (SWO)

Symbol	Description	Min	Nom	Max	Unit
F _{SWO}	SWO operation frequency	TBD	-	TBD	MHz
DC _{SWO}	SWO clock duty cycle	-	TBD	-	%
T _{RISE}	SWO rise time	-	TBD	-	TBD
T _{FALL}	SWO fall time	-	TBD	-	TBD
T _{SETUP}	SWO setup time	-	TBD	-	TBD
T _{HOLD}	SWO hold time	-	TBD	-	TBD

16.21 Device Errata and Design Considerations

16.21.1 Apollo Device Errata

The section describes silicon errata for the Apollo MCU device related to operating temperature range, IO master functionality, ADC, and the PDR threshold. The errata listed below apply to devices with the following part numbers: APOLLO512-KBR, APOLLO512-KCR, APOLLO256KBR, APOLLO256-KCR, APOLLO128-KBR, APOLLO128-KCR, APOLLO064-KBR, and APOLLO064-KCR.

16.21.1.1 Operating Temperature Range

16.21.1.1.1 Errata Details & Impact

Revision A3 Apollo MCU devices are only guaranteed to operate within the -10°C to 60°C temperature range. This range is less than the one shown in the Apollo MCU datasheet of -40°C to 85°C. The reduced range is required to guarantee operation as process and voltage vary in revision A3, in addition to temperature.

16.21.1.1.2 Workaround

Customers must use Apollo revision A3 within the reduced temperature range of -10°C to 60°C.

16.21.1.1.3 Resolution

A new silicon revision now in production (A4), expands the operating temperature range to the expected values of -40°C to 85°C.

16.21.1.2 IO Master Clock Pin

16.21.1.2.1 Errata Details & Impact

Apollo MCU devices support configuring each of the two IO Master peripherals in either I²C mode or SPI mode. When configured in either mode, the device IO masters may experience issues where a transaction is corrupted. The prevalence of this issue depends on a combination of normal device package variation, board trace inductance, capacitive loading of the SCL/SCLK line, and pad drive strength settings.

Under the right conditions, a second clock edge forms on the SCL/SCLK line that is read by the IO Master peripheral, which puts it in an undetermined state, corrupting a normal transaction.

This errata affects Apollo MCU revision A3, and A4.

16.21.1.2.2 Workaround

To prevent this issue across all possible board configurations, Ambiq Micro requires that customers place a 100 Ω resistor in series with the SCL/SCLK pin. This resistor should be placed close to the Apollo MCU device.

16.21.1.3 ADC OPMODE Supply Voltage Dependency

16.21.1.3.1 Errata Details & Impact

The Apollo ADC peripheral includes a mode controller state machine that manages time slot conversions, clock management of the block, and also the power state of its various analog components. The OPMODE bits in the ADC CFG register configure the sample rate mode to be used by the controller. The reset value for the OPMODE field is 0x0, which configures a sample rate less than or equal to 125 kSps. A setting of 0x02 configures the ADC for its highest sample rate of up to 800 kSps.

When the OPMODE bits configure the ADC for the highest sample rate, and the supply voltage of the device is less than 2.0 V, the ADC may not operate correctly.

This errata affects Apollo MCU revision A3, and A4.

16.21.1.3.2 Workaround

Customers must set the OPMODE bits to 0x0 if using the ADC peripheral on Apollo when the supply voltage is less than 2.0 V. This setting guarantees proper operation of the ADC, albeit at a lower sampling rate.

16.21.1.4 ADC Trigger 4 Affects Internal Voltage Rails

16.21.1.4.1 Errata Details & Impact

Apollo's ADC can be triggered from nine different sources, eight of which are GPIOs configured as external triggers. Using ADC external trigger 4 (GPIO41), can affect sampling of the VDD input to the ADC due to coupling of the rising edge of the trigger signal. This coupling results in an offset of the ADC measurements of VDD, where the resulting output code is lower than expected. This phenomenon only occurs when using ADC external trigger 4.

This errata affects Apollo MCU revision A3, and A4.

16.21.1.4.2 Workaround

Ambiq Micro recommends not using ADC external trigger 4. This functionality of GPIO41 will be deprecated in the next version of the Apollo datasheet.

16.21.1.5 POR Rising Slew Rate Limitation

16.21.1.5.1 Errata Details & Impact

Revision A3 Apollo devices require a rising slew rate to be 10 V/s or faster to guarantee that the MCU properly comes out of the reset condition. A voltage supply ramp rate less than 10 V/s may result in the MCU never coming out of reset.

16.21.1.5.2 Workaround

Customers using revision A3 must ensure a 10 V/s or higher ramp rate on the MCU input power supply pins.

16.21.1.5.3 Resolution

A new silicon revision now in production (A4) corrects this issue, removing any slew rate requirements for the input supply.

16.21.1.6 Falling POR Threshold At Cold Temperatures

16.21.1.6.1 Errata Details & Impact

Apollo has a falling edge POR (power-on reset) threshold set to approximately 1.72 V, which will safely reset the MCU state prior to its input power supply falling below the absolute minimum value, which would cause functional failures. While operating the ADC peripheral at cold temperatures, between -10°C and -40°C, it's possible for a system functional failure to occur above this 1.72 V threshold. Operating the ADC at cold temperatures can possibly cause the voltage of an internal supply to drop below 1.72 V, while the supply coming into the chip remains above the threshold.

This errata affects Apollo MCU revision A3, and A4.

16.21.1.6.2 Workaround

As a result, Ambiq Micro recommends using the BOD (brown-out detection) feature, with a threshold set to 2.1 V, to detect a low supply condition at cold temperatures, and disable the ADC in these situations. See Section 11.4 of the Apollo datasheet for more details on using the BOD monitor to detect brown-out events.

16.21.2 Key Design Guidelines

16.21.2.1 Input Supply Voltage and Rising POR Threshold

The Apollo device is designed to work with input power supplies between 1.8 V and 3.8 V, **after a power-on reset condition has been de-asserted**. However, to de-assert the power-on reset condition, a supply voltage higher than the POR threshold of ≥ 2.05 V must be applied to the device for at minimum 100 ms. After that condition is satisfied, system reset is de-asserted and normal MCU operation proceeds.

Typical power supply designs do not allow for a short 100 ms voltage pulse of ≥ 2.05 V, followed by a steady-state value of around 1.8 V. Therefore, Ambiq Micro recommends that most customers use a voltage supply at minimum of 2.1 V to power the Apollo MCU, unless such a voltage sequence is possible in their design.

16.21.2.2 ADC Channel 0 Input Leakage

The Apollo ADC supports 13 user-selectable input channels, eight of which are external GPIO inputs. ADC external inputs 1 through 7 have an input leakage current specification of typically 0.1 nA, with a maximum leakage of 20 nA. ADC external input 0 has a much higher input leakage current, typically 5 μ A, but as high as 10 μ A.

The additional leakage current on external input 0 can create a noticeable offset in ADC samples, depending on the output impedance of the signal source being measured. Customers using external input 0 must be aware of this additional leakage and account for it when selecting a signal source to be sampled, or use one of the other seven external inputs.

16.21.2.3 32.768 kHz Crystal Considerations

16.21.2.3.1 Crystal Load Capacitors and Frequency Accuracy

The Apollo MCU clock generator module includes a high-precision crystal-controlled oscillator that works in conjunction with an external 32.768 kHz tuning fork crystal. The crystal oscillator includes a distributed

digital calibration function, that can be used to calibrate the 16 kHz level of the internal MCU clock divider chain to within ± 1 ppm accuracy.

The crystal oscillator is designed to run without the use of external load capacitors, and use the digital calibration function to create a highly accurate clock for internal use. Not using external load capacitors significantly increases oscillator allowance, and allows for a reduction in crystal bias current, which results in lower power consumption. Lower power consumption is often most beneficial when the MCU is in deep sleep mode, but needs to keep the crystal oscillator running as an accurate time source for the RTC module.

16.21.2.3.2 Replacing Crystal with an XO or TCXO

Due to design requirements, some customers prefer to use an XO (external oscillator) device, or a TCXO (temperature-controlled external oscillator) as a replacement for the 32.768 kHz crystal. The Apollo MCU crystal oscillator was not originally designed to support these kinds of inputs but can be adapted to work with them.

The crystal-controlled oscillator circuit expects an input between 150 mV_{pp} and 250 mV_{pp} , with a nominal value of 200 mV_{pp} . Typically, XO-type devices have a much higher output voltage, and must be AC-coupled into the Apollo oscillator input. To do so, Ambiq Micro recommends using a capacitive divider between the XO/TCXO output and the XO pin of the device. The XI pin must not be left floating. Instead, a 1 nF capacitor should connect the XI pin to the MCU's ground. See Figure 61 below for an example of a circuit illustrating the recommended connection.

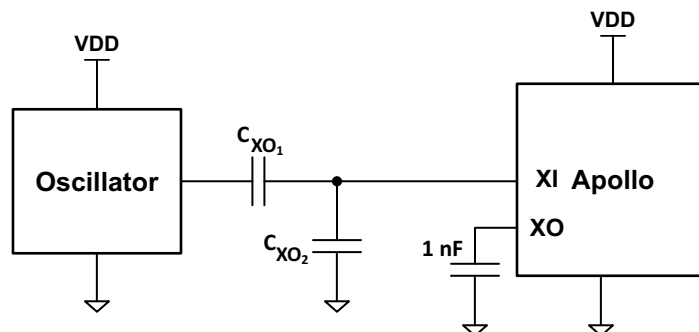


Figure 61. External Oscillator Example Connection

16.21.2.4 Updated DC-DC Converter Requirements

Apollo features two built-in Buck converters optimized for low-power environments, featuring ultra-low quiescent current and an integrated power switch. Therefore, only an external inductor and capacitor are needed for each converter to enable very high efficiency power input for the MCU core, SRAM, Flash memory, analog blocks, and digital logic.

To ensure best performance across process, voltage, and temperature, additional capacitance is needed at the VOUT1 Buck converter output to help reduce ripple. Compared to older Apollo example schematics, the VOUT1 output capacitor must be increased from $1\text{ }\mu\text{F}$ to $2\text{ }\mu\text{F}$. Using a standard capacitor value of $2.2\text{ }\mu\text{F}$ is also acceptable. See Figure 62 below for an updated Buck converter example schematic.

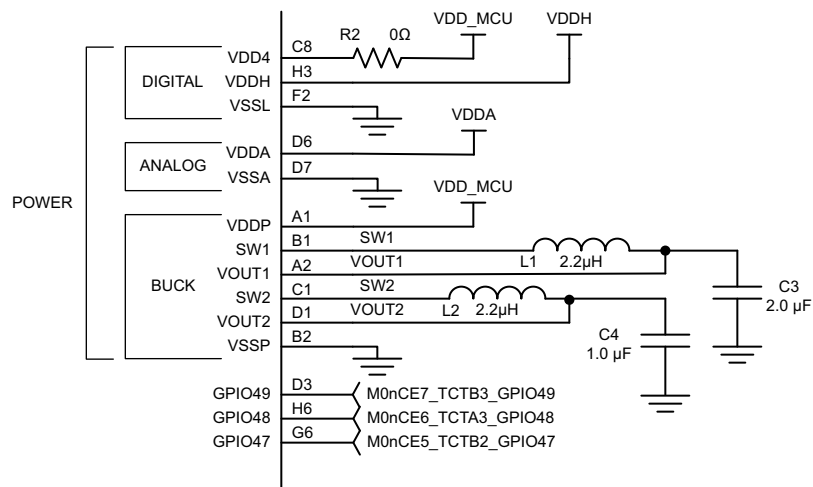
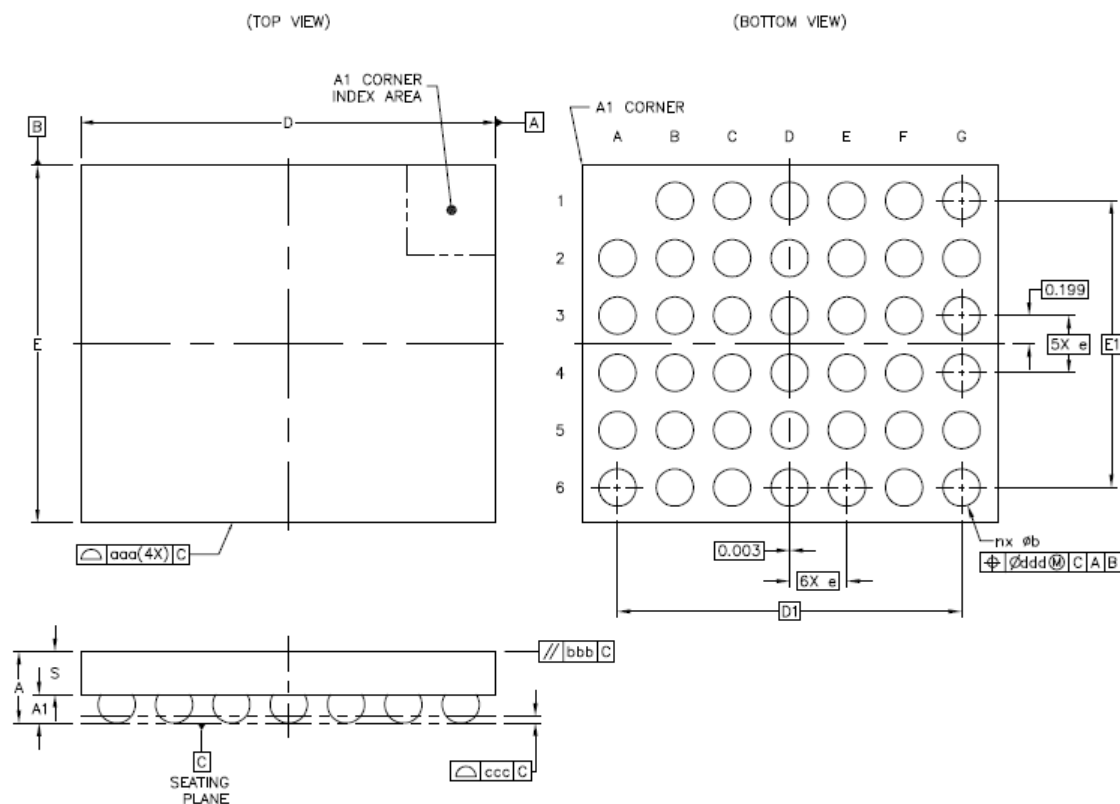


Figure 62. Updated Buck Converter Schematic, Showing C3 Value as 2.0 µF

17. Package Mechanical Information

17.1 CSP Package¹



	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

1. All dimensions in mm unless otherwise noted.

17.1.1 PCB land pattern and solder stencil

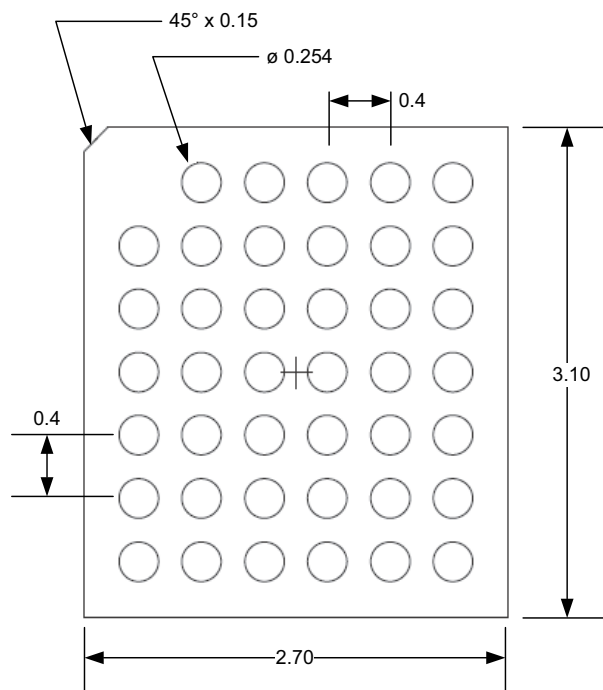


Figure 64. Example PCB Land Pattern for CSP Package

A 4 mil stencil is recommended.

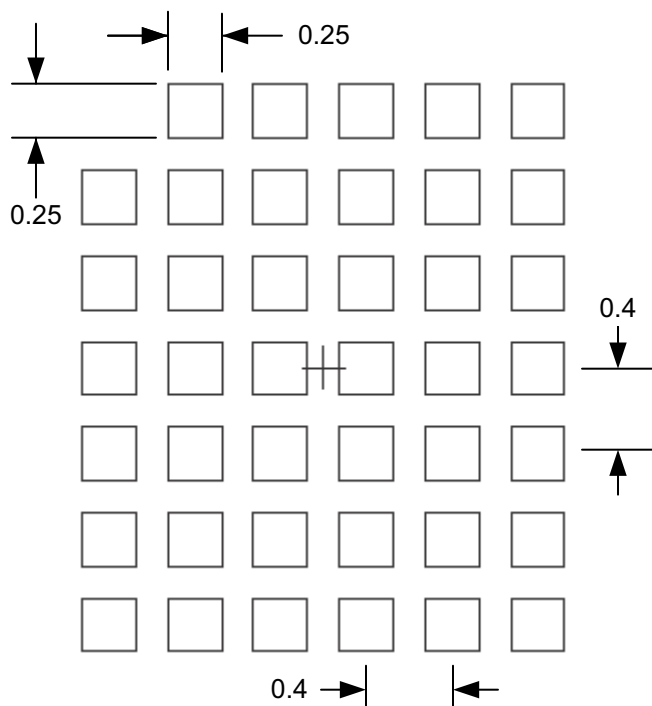
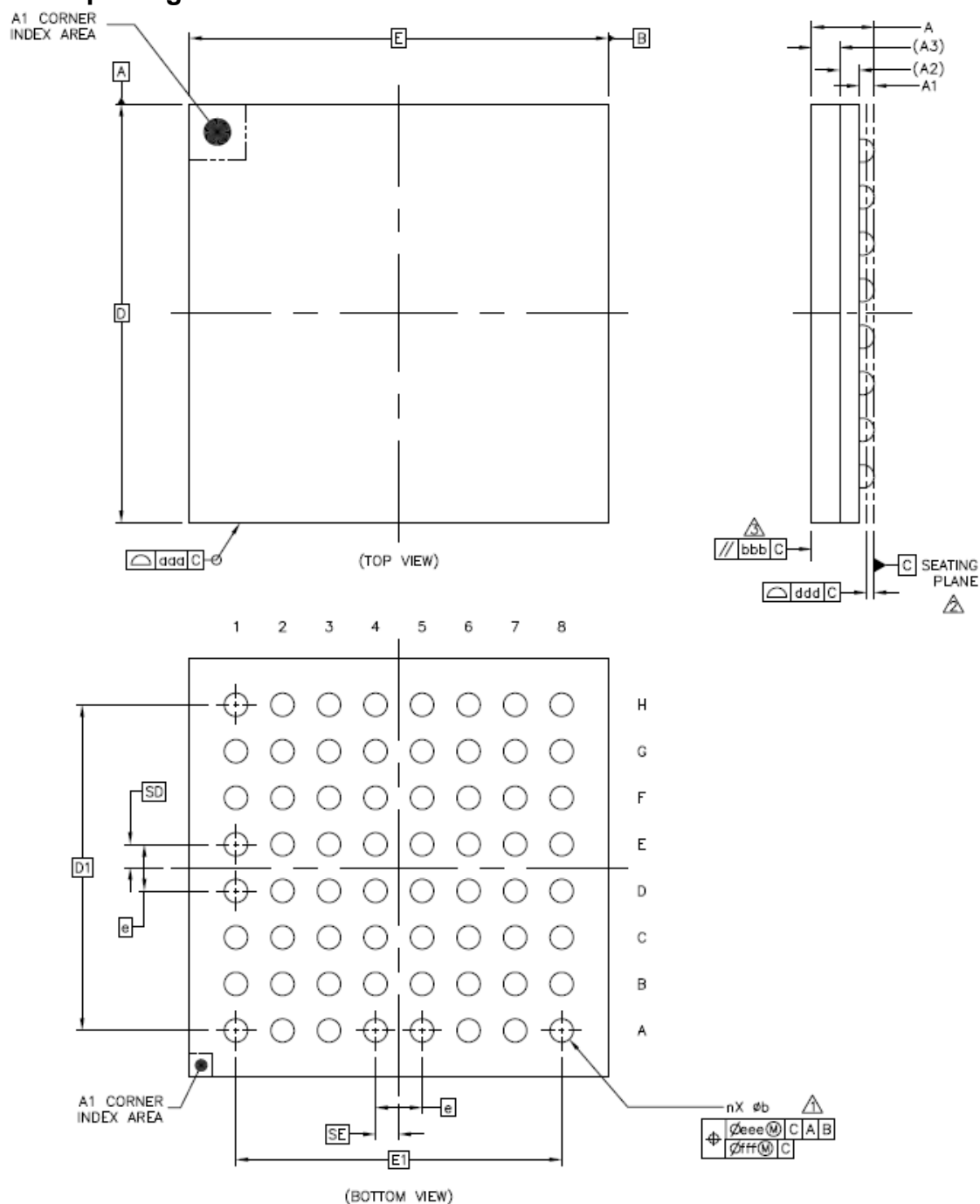


Figure 65. Example Solder Stencil Pattern for CSP Package

17.2 BGA package¹



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.

17.2.1 PCB land pattern and solder stencil

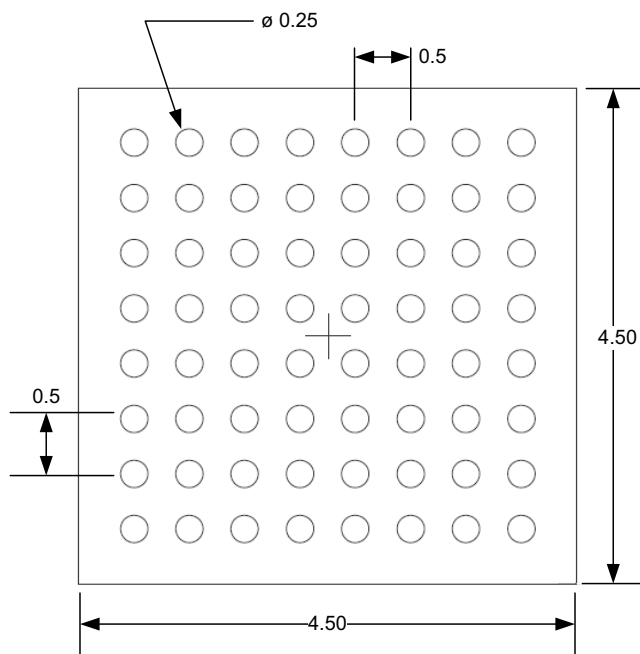


Figure 66. Example PCB Land Pattern for BGA Package

A 4 mil stencil is recommended.

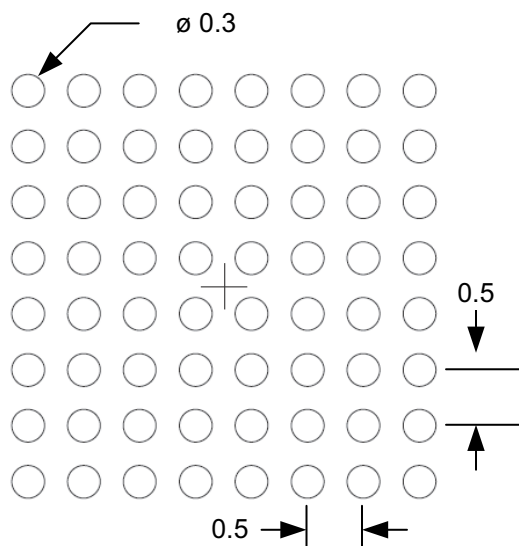


Figure 67. Example Solder Stencil Pattern for BGA Package

17.3 Reflow Profile

Figure 68 illustrates the reflow soldering requirements.

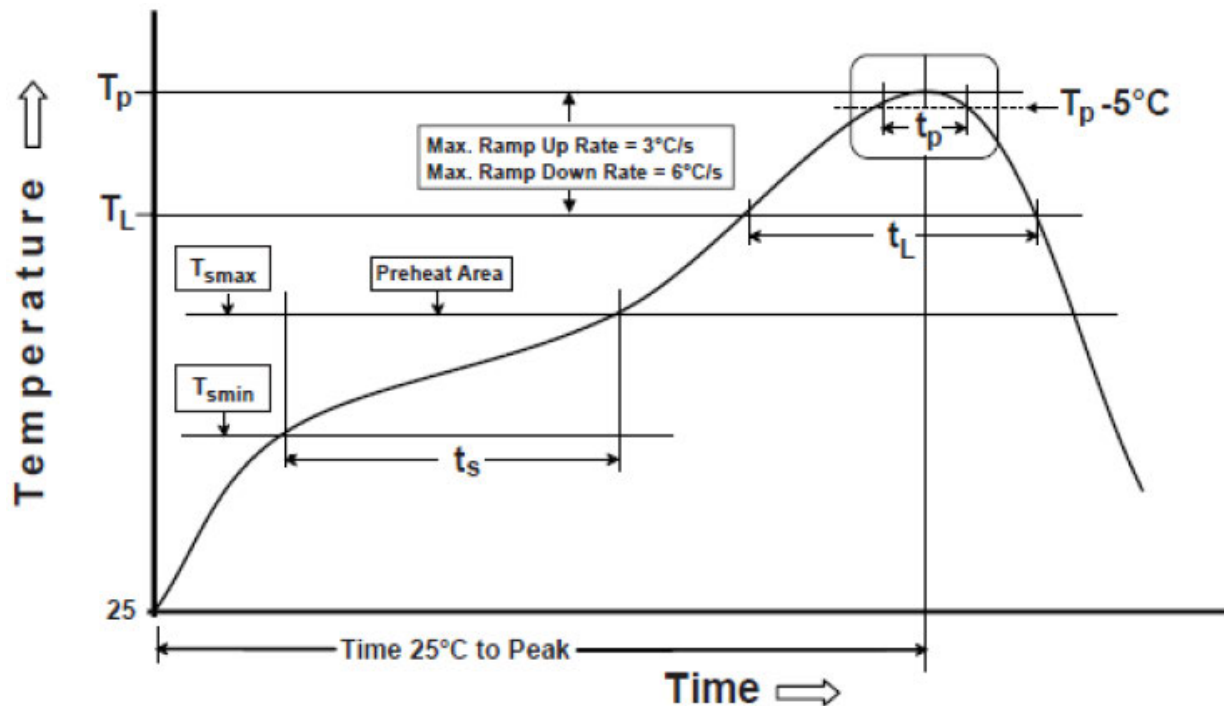


Figure 68. Reflow Soldering Diagram

Table 565: Reflow Soldering Requirements (Pb-free assembly)

Profile Feature	Requirement
Preheat/Soak	
Temperature Min (T_{smin})	150°C
Temperature Max (T_{smax})	200°C
Time (t_s) from (T_{smin} to T_{smax})	60-120 seconds
Ramp-up rate (T_L to T_p)	3°C/second max.
Liquidous temperature (T_L)	217°C
Time (t_L) maintained above T_L	60-150 seconds
Peak package body temperature (T_p)	260°C max.
Time (t_p) within 5°C of T_p	30 seconds max.
Ramp-down rate (T_p to T_L)	6°C/second max.
Time 25°C to peak temperature	8 minutes max.

Index

Numerics

10BIT	91
12/24 Hour Mode	243
3-wire mode	94

A

Access permissions	37
ACC_ISR	112
accumulation control	312
accumulation, automatic	312
ACK	91, 115
Acknowledge	91
Active Mode	37
ADC	33, 155, 309
ADC Configuration Register	315
ADC_DIV3	310
ADC_DIV3_LOAD	310
ADC_EXT0	310
ADC_EXT1	310
ADC_EXT2	310
ADC_EXT3	310
ADC_EXT4	310
ADC_EXT5	310
ADC_EXT6	310
ADC_EXT7	310
ADCREf	155
ADC_SWT	311
ADC_TEMP	310
ADC_TRIG0	310
ADC_TRIG1	310
ADC_TRIG2	311
ADC_TRIG3	311
ADC_TRIG5	311
ADC_TRIG6	311
ADC_TRIG7	311
ADC_VDD	310
ADC_VSS	310
AHB	37, 111
Alarm Registers	242
ALM interrupt	242
ALM100	242
AM08XX, see real-time clock	
AM18XX, see real-time clock	
AMBA	37
Analog Multiplexer Channel Selection	312
analog mux	310
analog-to-digital converter	33
APB	37
Apollo	32
Architecture Reference Manual	38
Area, Direct	112
ARM, see processor	
ARMv7	37

Attachment, debugger	38
Autoadjustment, HFRC	226
Automatic Sample Accumulation	312

B

Bandgap	311
battery life	32
BCD format	242
Buck Converter Charge Insertion	256
Buck Converters	351
Bus Not Busy	90, 115
bus, AMBA AHB	37
bus, AMBA APB	37
bus, DCode	37
bus, ICode	37
bus, System	37

C

Calibration, Distributed Digital	225
CALRC	225
CALXT	225
CLKOUT	224, 225, 227
Clock Generator Module	224
clock sources	33
clock, interface	88
CMPOUT	344
CMPR0	254, 255
CMPR1	255
comparator, window	309
CONT	93
Continuous	255
control, accumulation	312
converters, buck	351
core, see processor	
Cortex, see processor	
counter, 32-bit	256
CPHA	94, 97, 120
CPOL	94, 97, 120
CTRERR	242
CTS	154

D

Data Valid	90, 115
DCode	37
debug	38
Debug Interfaces	38
debugger attachment	38
Deep Sleep	225
Deep Sleep Mode	37, 38
Deep Slope Mode	120
Direct Area	112
Distributed Digital Calibration	225

E

EG_IOSLAVE_FIFOPTR_FIFOSIZ	113
event	33
event, wakeup	37

F

Fast Mode Plus	114
fault handler, Memanage	37
Faulting Address Trapping Hardware	39
FIFO	112, 309
Area Functions	113
FIFOBASE	112
FIFOCTR	113
FIFOMAX	112
FIFOPTR	113
FIFOREM	89
FIFORTH	90
FIFOSIZ	89, 113
FIFOUPD	114
FIFOWTHR	90
flash	33

G

GPIO	142
GPIO and Pad Configuration Module	139
GPIOA_IER	142
GPIOA_ISR	142
GPIOA_WCR	142
GPIOB_IER	143
GPIOB_ISR	143
GPIOB_WCR	143
GPIOB_WSR	142, 143
GPIOEN	142
GPIOENA	142
GPIOENB	142
GPIOOnINCFG	142, 145
GPIOOn_INT	145
GPIOOnINTD	145
GPIOOnINTP	142
GPIOOnOUTCFG	144
GPIOORD	145
GPIOORDA	142
GPIOORDB	142
GPIOWT	142, 143
GPIOWTA	142
GPIOWTB	142
GPIOWTCA	142
GPIOWTCB	142
GPIOWTSA	142
GPIOWTB	142

H

Hardware, Fault Address Trapping	39
HFADJ Register	235
HFADJCK	227
HFRC	226, 310
HFRC Autoadjustment	226
HFRCDEL	228
HFXTADJ	226
High Frequency RC Oscillator	226
HR1224	243

I

I2C	
10-bit addressing	116
7-bit addressing	116
ADDRESS	91
Address	116
Command	91
FIFO	89
I2CADDR	114, 116
IO Master 0	145
IO Master 0 Loopback	150
IO Master 1	146
IO Master 1 Loopback	150
IO Slave	149
master	88
Multi-master Arbitration	94
Normal Read	92
Normal Write	92
Offset Address	92, 116
Raw Read	93
Raw Write	93
Read	117
receiver	90
SCL	90
SDA	90
Slave	111, 114
transmitter	90
Write	117
I2C/SPI Master	224
I2C/SPI Master Module	88
ICode	37
IE bit	253
Instrumentation Trace Macrocell	38
Instrumentation Trace Module	33
interfaces, debug	38
interrupt	33, 34
ALM	242
IOINT	114
RDERR	114
Interrupts	
Vector Table	34
IO Slave Interrupt	150
IOINT	114
IOREAD	114
ITM, see Instrumentation Trace Macrocell	
L	
LDO	351
LFRC	225, 228
life, battery	32
Loopback	150
Low-Power Consumption Modes	33
M	
Managed Conversion Slots	311
map, memory	35

Master Module , I2C/SPI	88	PAD9PULL	146, 147
MCU, see Apollo<\$noage	32	PADKEY	142
MCU_CTRL	228	PADnFNCSEL	139–??, 140, 142
MemManage	37	PADnINPEN	145, 148, 153
memory	33	PADnPULL	140, 145, 148, 153
LRAM	89, 111	PADnRSEL	145
RAM	33	PADnSTRNG	144
SRAM	37	Peripheral Device Memory Map	36
memory map	35	PINCFG	94
peripheral device	36	power	32
Memory Protection Unit	36	power modes	37
MISO	94	Power Optimization	256
mode		processor	33
Active	37	Protected Memory System Architecture	36
Deep Sleep	38, 225	Protection regions	36
Sleep	38	PSEL	344
mode, Deep Sleep	37	PTAT	344
Module, ADC and Temperature Sensor	309	PWDKEY	344
module, PINCFG	94	R	
MOSI	94	rate, sampling	309
MPU, see Memory Protection Unit		RDERR	114
mux, analog	310	Real Time Clock Module	227
N		real-time clock	33
NAK	116	REG_CLK_GEN_ALMLOW_ALM100	242
Nested Vectored Interrupt Controller	34	REG_CLK_GEN_ALMLOW_ALMHR	242
NSEL	344	REG_CLK_GEN_ALMLOW_ALMMIN	242
NVIC, see Nested Vectored Interrupt Controller		REG_CLK_GEN_ALMLOW_ALMSEC	242
O		REG_CLK_GEN_ALMUP_ALMDATE	242
OPER	92	REG_CLK_GEN_ALMUP_ALMMO	242
oscillator		REG_CLK_GEN_ALMUP_ALMWKDY	242
High Frequency RC	226	REG_CLKGEN_CALXT	225
high frequency RC	33, 224, 310	REG_CLKGEN_CCTRL_CKEN	227
low frequency RC	33, 224, 225	REG_CLKGEN_CCTRL_CKSEL	227
RC	33	REG_CLK_GEN_CTRLOW_CTR100	241
XTAL	33, 224, 225	REG_CLK_GEN_CTRLOW_CTRHR	241
XTAL Failure Detection	228	REG_CLK_GEN_CTRLOW_CTRMIN	241
OUTDATSEL	143	REG_CLK_GEN_CTRLOW_CTRSEC	241
OUTENSEL	143, 144, 145	REG_CLK_GEN_CTRUP_CB	241
P		REG_CLK_GEN_CTRUP_CEB	243
PAD10PULL	147	REG_CLK_GEN_CTRUP_CTRDATE	241
PAD16FNCSEL	155	REG_CLK_GEN_CTRUP_CTRERR	242
PAD20FNCSEL	156	REG_CLK_GEN_CTRUP_CTRMO	241
PAD20INPEN	156	REG_CLK_GEN_CTRUP_CTRWKDY	241
PAD21INPEN	156	REG_CLK_GEN_CTRUP_CTRYR	241
PAD5INPEN	145, 146	REG_CLKGEN_HFADJ_HFADJEN	227
PAD5PULL	145, 146	REG_CLKGEN_HFTUNERB	227
PAD5RSEL	145	REG_CLKGEN_OCTRL_OSEL	225, 228
PAD6INPEN	145, 146	REG_CLK_GEN_RTCCTL_HR1224	243
PAD6PULL	145, 146	REG_CLK_GEN_RTCCTL_RPT	242
PAD6RSEL	145	REG_CLK_GEN_RTCCTL_RSTOP	242
PAD7PULL	146	REG_CLK_GEN_RTCCTL_WRTC	242
PAD8INPEN	146, 147	REG_CLKGEN_STATUS_OMODE	228
PAD8PULL	146, 147	REG_CLKGEN_STATUS_OSCF	228
PAD9INPEN	146, 147	REG_CTIMER_CMPR0	253

REG_CTIMER_CMPR0/1	252	Serial Wire Debugger	33
REG_CTIMER_CTCTRLx_CTRLINKx	256	Single Count	253
REG_CTIMER_CTCTRLx_TMRxyFN	252	Single Pulse	254
REG_CTIMER_TMRxyCLR	253	Sleep Mode	38, 120
REG_CTIMER_TMRxyEN	254	SLEEPONEXIT	38
REG_CTIMER_TMRxyIE	254	SLOTS	311
REG_CTIMER_TMRxyPE	253, 254	Slots, Mananged Conversion	311
REG_CTIMER_TMRxyPOL	254	SPHA	97
REG_GPIO_GPIOCFGy	142	SPI	94
REG_GPIO_GPIOyCFG_GPIOOnOUTCFG	142	3-wire	119
REG_GPIO_PADKEY	139	CHANNEL	94
REG_GPIO_PADREG	139	CMD Register	94
REG_GPIO_PADREGy_PAD11PWRDN	139	Complex Operations	96
REG_GPIO_PADREGy_PAD20PULL	139	frequency	88
REG_GPIO_PADREGy_PAD3PWRUP	139	IO Master 0 3-wire	148
REG_GPIO_PADREGy_PAD4PWRUP	139	IO Master 0 3-wire Loopback	151
REG_GPIO_PADREGy_PADnFNCSEL	139	IO Master 0 4-wire	146
REG_GPIO_PADREGy_PADnINPEN	139	IO Master 0 4-wire Loopback	151
REG_GPIO_PADREGy_PADnPULL	139	IO Master 1 3-wire	149
REG_GPIO_PADREGy_PADnSTRNG	139	IO Master 1 3-wire Loopback	152
REG_IOMSTRn_CLKCFG_DIV3	88	IO Master 1 4-wire	147
REG_IOMSTRn_CLKCFG_DIVEN	89	IO Master 1 4-wire Loopback	151
REG_IOMSTRn_CLKCFG_FSEL	88	IO Slave 3-wire	149
REG_IOMSTRn_CLKCFG_LOWPER	89	IO Slave 4-wire	149
REG_IOMSTRn_CLKCFG_TOTPER	88	master	88
REG_IOMSTRn_CMD	89	Normal Read	95
REG_IOMSTRn_FIFOPTR_FIFOREM	89	Normal Write	94
REG_IOMSTRn_FIFOPTR_FIFOSIZ	89	OPER	94
REG_IOMSTRn_FIFOTHR_FIFOWTHR	89	Phase	97
REG_IOMSTRn_IOMCFG_SPHA	94	Polarity	97
REG_IOMSTRn_IOMCFG_SPOL	94	Raw Read	96
REG_IOSLAVE_FIFOCFG_FIFOBASE	112	Raw Write	95
REG_IOSLAVE_FIFOCFG_FIFOMAX	112	Read	119
REG_IOSLAVE_FIFOCFG_ROBASE	113	Slave	111, 118
REG_IOSLAVE_FIFOCTR	113	slave	118
REG_IOSLAVE_FIFOPTR_FIFOPTR	113	Slave Addressing	94
REG_IOSLAVE_FUPD_FIFOUPTD	114	Write	118
REG_IOSLAVE_FUPD_IOREAD	114	SPOL	97, 120
REG_IOSLAVE_IOSCFG_I2CADDR	114	START	115
REG_IOSLAVE_IOSCFG_LSB	120	Start Data Transfer	90, 115
REG_IOSLAVE_PRENC	113	STOP	93, 115
REG_MCU_CTRL_HFRC_HFRCDEL	228	STOP condition	90
Repeated Count	253	Stop Data Transfer	90, 115
Repeated Pulse	254	Successive Approximation Register ...	33, 309
Reset Module	286	SWD, see Serial Wire Debug	
RESTART	115	SWD, see Serial Wire Debugger	
RSTn	286	SWDCK	156
RTC, see real-time clock		SWDIO	156
RTS	154	SYSRESETREQn	286
S		System Control Register	37
sampling rate	309	T	
SAR, see Successive Approximation Register		Temperature Sensor	309
SCR, see System Control Register		TMRWCR	253, 254
Serial Wire Debug	38	TPIU, see Trace Port Interface Unit	

Trace Port Interface Unit	33, 38
Track and Hold Time	312
TRIGSEL	311
TX	153
U	
UART	153, 226
CTS	154
RTS	154
RX	153
UPLNGTH	94
V	
VCOMP, see Voltage Comparator	
VEXT1	344
VEXT2	344
Voltage Comparator	156, 344
Voltage Regulator Module	351
VREFEXT1	344
VREFEXT2	344
VREFEXT3	344
VREFINT	344
VTEMP	344
W	
Wait-For-Interrupt	38
Wakeup	120
wake-up	33
Wake-Up Interrupt Controller	37
Watchdog Timer	280
WC bit	253
WDTCFG	280
WFI, see Wait-For-Interrupt	
WIC, see wake-up	
Window comparator	309
Window Comparisons	312
X	
XT	225, 227
XT Oscillator	225

18. Ordering Information

Table 566: Ordering Information

Orderable Part Number	Flash	RAM	Package ^{a,b}	Packing	Temperature Range ^c	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	4Q15
APOLLO256-KBR	256 KB	32 KB	64-pin BGA	Tape and Reel	-40 to +85°C	4Q15
APOLLO128-KCR	128 KB	32 KB	41-pin WLCSP	Tape and Reel	-40 to +85°C	4Q15
APOLLO128-KBR	128 KB	32 KB	64-pin BGA	Tape and Reel	-40 to +85°C	4Q15
APOLLO064-KCR	64 KB	16 KB	41-pin WLCSP	Tape and Reel	-40 to +85°C	4Q15
APOLLO064-KBR	64 KB	16 KB	64-pin BGA	Tape and Reel	-40 to +85°C	4Q15

a. For 41-pin WLCSP, minimum order quantity of 5,000 pieces.

For 64-pin BGA, minimum order quantity of 4,000 pieces.

b. 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.

c. Temperature Range is guaranteed -40 to +85°C by design. Current production test limits are -10 to +60°C

19. Document Revision History

Table 567: 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.

Contact Information

Address	Ambiq Micro, Inc. 6500 River Place Blvd. Building 7, Suite 200 Austin, TX 78730-1156
Phone	+1 (512) 394-8542
Website	http://www.ambiqmicro.com/
General Information	info@ambiqmicro.com
Sales	sales@ambiqmicro.com
Technical Support	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.