



# ADuCRF101 USER GUIDE

**Note: This is a draft document and is not a final representation of the ADuCRF101.**

Version PrA, November 2010  
Internal Report No.1057

# **DISCLAIMER**

This document contains Analog Devices confidential information.

Analog Devices, Inc. reserves the right to change this product without prior notice.

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use; nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under the patent rights of Analog Devices, Inc.

Trademarks and registered trademarks are the property of their respective owners.

# CONTENTS

DISCLAIMER .....	2
CONTENTS .....	3
List of Tables .....	7
List of Figures .....	15
Introduction .....	17
Purpose of Document.....	17
Revision History .....	17
Terminology.....	18
Architectural Overview .....	19
Product Features .....	19
ADuCRF101 Block Diagram .....	20
Memory Organisation .....	20
System Clocks .....	23
System Clock Features .....	23
System Clocks Operation .....	24
16MHz Oscillator Trim .....	25
System Clocks Memory Mapped Registers .....	27
Power Management Unit.....	32
Power Management Unit Features .....	32
Power Management Unit Block Diagram .....	33
Power Management Unit LDO switching .....	33
Power Management Unit Power Modes Operation .....	34
Power Modes Memory Mapped Registers .....	36
Cortex-M3 Core .....	37
Cortex-M3 Core Features .....	37
Cortex-M3 Core Functional Description .....	39
Exceptions & Interrupts .....	40
Cortex-M3 and Fault Management .....	40
Nested Vectored Interrupt Controller .....	41
External Interrupt Configuration .....	43
Interrupt Memory Mapped Registers .....	44
DMA Controller .....	50
DMA Features .....	50
DMA Operation.....	51
DMA Memory Mapped Registers.....	60
Flash Controller.....	78
Flash Controller Features .....	78
Flash Memory Organization.....	79

Writing to Flash/EE Memory .....	80
Erasing Flash/EE Memory .....	81
Flash Protection .....	81
Flash Controller Failure Analysis Key .....	82
Flash Integrity Signature feature .....	83
Integrity of the Kernel .....	83
Abort Using Interrupts.....	84
Flash Controller Performance and Command Duration .....	84
Flash Controller Memory Mapped Registers.....	85
Reset .....	93
Reset Operation .....	93
Reset Memory Mapped Registers .....	94
UHF Transceiver.....	95
UHF Transceiver Features .....	95
UHF Transceiver Simplified Block Diagram .....	95
Radio Control .....	96
Packet Mode .....	111
Data Whitening.....	120
8b/10b Encoding .....	121
Sport Mode.....	121
Interrupt Generation .....	127
UHF Transceiver Memory Map.....	131
UHF Transceiver SPI Interface.....	134
Low Power Modes .....	144
Downloadable Firmware Modules.....	154
Transceiver Radio Blocks .....	157
Crystal Oscillator .....	160
Recommended Receiver Settings for 2FSK/GFSK/MSK/GMSK .....	173
Recommended Receiver Settings for OOK .....	176
Peripheral Features .....	177
BBRAM Register Maps.....	179
Modem Configuration Register Map .....	180
BBRAM Register Description.....	182
MCR Register Description .....	200
Digital I/Os .....	212
Digital I/Os Functionality .....	212
Digital Port Multiplex.....	212
Inter-die Connectivity .....	215
Access to Internal Signals .....	216
Digital I/O Block Diagram.....	217
Digital I/O Operation .....	217

GPIO Memory Mapped Registers .....	219
I <sup>2</sup> C Serial Interface .....	225
I <sup>2</sup> C Functionality .....	225
I <sup>2</sup> C Operation .....	225
I <sup>2</sup> C Memory Mapped Registers .....	231
Serial Peripheral Interfaces .....	244
SPI Features .....	244
SPI Operation .....	245
SPI Memory Mapped Registers .....	251
UART Serial Interface .....	258
UART Features .....	258
UART Operation .....	258
UART Memory Mapped Registers .....	260
General Purpose Timers .....	268
General Purpose Timers Features .....	268
General Purpose Timers Block Diagram .....	269
General Purpose Timers Operation .....	269
General Purpose Timers Memory Mapped Registers .....	271
Wake Up Timer .....	275
Wake Up Timer Features .....	275
Wake Up Timer Block Diagram .....	275
Wake Up Timer Operation .....	276
Wake Up Timer Memory Mapped Registers .....	277
Watchdog Timer .....	284
Watchdog Timer Features .....	284
Watchdog Timer Block Diagram .....	284
Watchdog Timer Operation .....	284
Watchdog Timer Memory Mapped Registers .....	285
PWM .....	288
PWM Features .....	288
Operation in Standard Mode .....	288
PWM Memory Mapped Registers .....	291
Description of H-bridge Mode .....	294
ADC Circuit Overview .....	295
ADC Circuit Features .....	295
ADC Top Block Level Diagram .....	295
ADC Initialization .....	297
ADC .....	298
Interrupt Generation .....	298
ADC Calibration .....	298
ADC Memory Mapped Registers .....	299

Power Supply Support Circuits.....	303
Power Supply Support Circuits Features .....	303
Low Drop Out Regulators .....	303
Power-On-Reset.....	303
Power Supply Monitor .....	304
Power Supply Monitor Memory Mapped Registers .....	304
Hardware Design Considerations.....	305
ADuCRF101 Pin Configuration.....	305
Supplies and Grounds .....	308
Clocks .....	310
Serial Wire Debug Interface.....	311
PA/LNA Matching .....	312

# List of Tables

Table 1: ADuCRF101 Memory Map Register Summary .....	22
Table 2: Clock Control Memory Mapped Registers Address Table .....	27
Table 3: CLKCON Register Bit Description.....	27
Table 4: XOSCCON Register Bit Description.....	28
Table 5: HF Oscillator Trim Registers Address Table .....	28
Table 6: HFTSTA Register Bit Description.....	29
Table 7: HFTCON Register Bit Description.....	29
Table 8: HFTMAX Register Bit Description.....	30
Table 9: HFTMIN Register Bit Description .....	30
Table 10: HFTTRM Register Bit Description .....	30
Table 11: HFTXMAX Register Bit Description.....	30
Table 12: HFTXVAL Register Bit Description.....	31
Table 13: HFTUMAX Register Bit Description .....	31
Table 14: HFTUVAL Register Bit Description .....	31
Table 15: Power Modes Summary.....	35
Table 16: System Power Mode Summary.....	35
Table 18: PWRMOD Register Bit Description .....	36
Table 19: PWRKEY Register Bit Description .....	36
Table 20: List of System Exceptions.....	40
Table 21: Interrupt Vector Table .....	41
Table 22: NVIC Registers .....	43
Table 23: Interrupt Detection Unit Memory Mapped Registers Address.....	44
Table 24: EI0CFG Register Bit Description.....	45
Table 25: EI1CFG Register Bit Description.....	46
Table 26: EI2CFG Register Bit Description.....	47
Table 27: EICLR Register Bit Description .....	48
Table 28: NMICLR Register Bit Description.....	49
Table 29: DMA Channel Assignment.....	50
Table 30: Channel Control Data Structure .....	52
Table 33 CHNL_CFG For Primary Data Structure in Peripheral Scatter Gather Mode.....	59
Table 34: DMA Controller Memory Mapped Registers Address Table .....	60
Table 35: DMASTA Register Bit Description.....	61
Table 36: DMACFG Register Bit Description .....	61
Table 37: DMAPDBPTR Register Bit Description .....	62
Table 38: DMAADBPTR Register Bit Description .....	62
Table 39: DMASWREQ Register Bit Description .....	63
Table 40: DMARMSKSET Register Bit Description.....	64
Table 41: DMARMSKCLR Register Bit Description .....	66
Table 42: DMAENSET Register Bit Description.....	67

Table 43: DMAENCLR Register Bit Description.....	68
Table 44: DMAALTSET Register Bit Description .....	69
Table 45: DMAALTCLR Register Bit Description .....	71
Table 46: DMAPRISET Register Bit Description .....	72
Table 47: DMAPRICLR Register Bit Description.....	74
Table 48: DMAERRCLR Register Bit Description .....	75
Table 53: DMAPERID3 Register Bit Description .....	77
Table 54: DMAPCELLID0 Register Bit Description .....	77
Table 55: DMAPCELLID1 Register Bit Description .....	77
Table 56: DMAPCELLID2 Register Bit Description .....	77
Table 57: DMAPCELLID3 Register Bit Description .....	77
Table 58: Silicon Identification Register Bit Description .....	80
Table 59: Flash Controller Memory Mapped Registers Address Table .....	85
Table 60: FEESTA Register Bit Description.....	86
Table 62: FEECMD Register Bit Description.....	87
Table 63: Flash Controller Commands (FEECMD[3:0]) .....	88
Table 64: FEEADR0L Register Bit Description .....	89
Table 65: FEEADR0H Register Bit Description.....	89
Table 66: FEEADR1L Register Bit Description .....	89
Table 67: FEEADR1H Register Bit Description.....	89
Table 68: FEEKEY Register Bit Description.....	90
Table 69: FEEPROL Register Bit Description .....	90
Table 70: FEEPROH Register Bit Description .....	90
Table 71: FEESIGL Register Bit Description.....	90
Table 72: FEESIGH Register Bit Description .....	90
Table 73: FEECON1 Register Bit Description.....	91
Table 74: FEEADRAL Register Bit Description .....	91
Table 75: FEEADRAH Register Bit Description .....	91
Table 77: FEEAENX Register Bit Description .....	92
Table 78: Device Reset Implications.....	93
Table 79: Reset Memory Mapped Register Address Table.....	94
Table 80: RSTSTA/CLR Register Bit Description .....	94
Table 81: Sleep Modes Current Consumption .....	97
Table 82: Radio Controller Commands.....	101
Table 83: UHF Transceiver Command Execution Times and State Transition Times That Are Not Related to PHY_TX or PHY_RX .....	108
Table 84: UHF Transceiver State Transition Times Related to PHY_TX and PHY_RX.....	108
Table 85: Packet Structure Overview .....	111
Table 86: Preamble Detection Tolerance (PREAMBLE_MATCH, Address 0x11B).....	112
Table 87: Sync Word Programming Examples .....	114
Table 88: Sync Word Detection Tolerance .....	114



Table 89: Address Check Register Setup .....	117
Table 90: Example Address Check Configuration.....	118
Table 91: CRC Setup (prog_crc_en = 1) .....	119
Table 92: Example: Programming of crc_poly_0 and crc_poly_1.....	119
Table 93: SPORT Mode Setup .....	123
Table 94: GPIO Functionality in Sport Mode.....	124
Table 95: Structure of the Interrupt Mask Register 0.....	128
Table 96: Structure of the Interrupt Mask Register 1.....	129
Table 97: Structure of Interrupt Source Register 0.....	130
Table 98: Structure of Interrupt Source Register 1.....	130
Table 99: SPI Status Word .....	137
Table 100: Summary of SPI Memory Access Commands.....	139
Table 101: Overview of SPI Memory Access Commands.....	143
Table 102: Settings for Low Power Mode .....	145
Table 104: Automatic Synthesizer Bandwidth Selections .....	158
Table 105: PA Ramp Rate Settings .....	162
Table 106: AGC Gain Stages .....	164
Table 107: AGC Operation .....	164
Table 108. Summary of RSSI Measurement Methods .....	165
Table 109. Gain Mode Correction for 2FSK/GFSK/MSK/GMSK RSSI .....	166
Table 110. Gain Mode Correction for OOK RSSI.....	167
Table 111: Setting DISCRIM_PHASE[1:0].....	169
Table 112: AFC_LOCK_MODE Register Settings .....	170
Table 113: Maximum AFC Pull in Range .....	171
Table 114. Example Static Register Fix for AGC Settings.....	173
Table 115: Summary of AGC, AFC, Preamble Length, and Sync Word Error Tolerance for 2FSK/GFSK/MSK/GMSK.....	175
Table 116: Summary of Settings for AGC, AFC, and Preamble Length in OOK Demodulation .....	176
Table 117: Transmit Test Modes .....	178
Table 118. Product Code and Silicon Revision Code.....	178
Table 119: Battery Backup Memory (BBRAM).....	179
Table 120. Modem Configuration Memory (MCR) .....	180
Table 121. Packet RAM Memory .....	181
Table 122: 0x100 INTERRUPT_MASK_0.....	182
Table 123: 0x101 INTERRUPT_MASK_1.....	183
Table 124: 0x102 NUMBER_OF_WAKEUPS_0 .....	183
Table 125: 0x103 NUMBER_OF_WAKEUPS_1 .....	183
Table 126: 0x104 NUMBER_OF_WAKEUPS_IRQ_THRESHOLD_0 .....	183
Table 127: 0x105 NUMBER_OF_WAKEUPS_IRQ_THRESHOLD_1 .....	184
Table 128: 0x106 RX_DWELL_TIME .....	184

Table 129: 0x107 PARMTIME_DIVIDER .....	184
Table 130: 0x108 SWM_RSSI_THRESH .....	184
Table 131: 0x109 CHANNEL_FREQ_0 .....	184
Table 132: 0x10A CHANNEL_FREQ_1 .....	185
Table 133: 0x10B CHANNEL_FREQ_2 .....	185
Table 134: 0x10C RADIO_CFG_0 .....	185
Table 135: 0x10D RADIO_CFG_1 .....	185
Table 136: 0x10E RADIO_CFG_2 .....	185
Table 137: 0x10F RADIO_CFG_3 .....	186
Table 138: 0x110 RADIO_CFG_4 .....	187
Table 139: 0x111 RADIO_CFG_5 .....	187
Table 140: 0x112 RADIO_CFG_6 .....	187
Table 141: 0x113 RADIO_CFG_7 .....	188
Table 142: 0x114 RADIO_CFG_8 .....	189
Table 143: 0x115 RADIO_CFG_9 .....	190
Table 144: 0x116 RADIO_CFG_10 .....	191
Table 145: 0x117 RADIO_CFG_11 .....	191
Table 146: 0x118 IMAGE_REJECT_CAL_PHASE .....	192
Table 147: 0x119 IMAGE_REJECT_CAL_AMPLITUDE .....	192
Table 148: 0x11A MODE_CONTROL .....	193
Table 149: 0x11B PREAMBLE_MATCH .....	194
Table 150: 0x11C SYMBOL_MODE .....	194
Table 151: 0x11D PREAMBLE_LEN .....	195
Table 152: 0x11E CRC_POLY_0 .....	195
Table 153: 0x11F CRC_POLY_1 .....	195
Table 154: 0x120 SYNC_CONTROL .....	195
Table 155: 0x121 SYNC_BYTE_0 .....	196
Table 156: 0x122 SYNC_BYTE_1 .....	196
Table 157: 0x123 SYNC_BYTE_2 .....	196
Table 158: 0x124 TX_BASE_ADR .....	196
Table 159: 0x125 RX_BASE_ADR .....	196
Table 160: 0x126 PACKET_LENGTH_CONTROL .....	197
Table 161: 0x127 PACKET_LENGTH_MAX .....	198
Table 162: 0x128 STATIC_REG_FIX .....	198
Table 163: 0x129 ADDRESS_MATCH_OFFSET .....	198
Table 164: 0x12A ADDRESS_LENGTH .....	199
Table 165: 0x12B to 0x13D ADDRESS_FILTERING .....	199
Table 166: 0x13E RX_SYNTH_LOCK_TIME .....	199
Table 167: 0x13F TX_SYNTH_LOCK_TIME .....	199
Table 168: 0x307 PA_LEVEL_MCR .....	200
Table 169: 0x30C WUC_CONFIG_HIGH .....	200

Table 170: 0x30D WUC_CONFIG_LOW .....	201
Table 171: 0x30E WUC_VALUE_HIGH .....	201
Table 172: 0x30F WUC_VALUE_LOW .....	201
Table 173: 0x310 WUC_FLAG_RESET .....	202
Table 174: 0x311 WUC_STATUS .....	202
Table 175: 0x312 RSSI_READBACK .....	202
Table 176: 0x315 MAX_AFC_RANGE .....	202
Table 177: 0x319 IMAGE_REJECT_CAL_CONFIG .....	203
Table 178: 0x322 CHIP_SHUTDOWN .....	203
Table 179: 0x324 POWERDOWN_RX .....	203
Table 180: 0x325 POWERDOWN_AUX .....	204
Table 181: 0x327 ADC_READBACK_HIGH .....	204
Table 182: 0x328 ADC_READBACK_LOW .....	204
Table 184: 0x32E EXT_UC_CLK_DIVIDE .....	204
Table 185: 0x32F AGC_CLOCK_DIVIDE .....	205
Table 186: 0x336 INTERRUPT_SOURCE_0 .....	205
Table 187: 0x337 INTERRUPT_SOURCE_1 .....	206
Table 188: 0x338 CALIBRATION_CONTROL .....	206
Table 189: 0x339 CALIBRATION_STATUS .....	206
Table 191: 0x346 RXBB_CAL_CALWRD_OVERWRITE .....	207
Table 192: 0x359 ADC_CONFIG_LOW .....	207
Table 193: 0x35A ADC_CONFIG_HIGH .....	207
Table 194: 0x35B AGC_OOK_CONTROL .....	208
Table 195: 0x35C AGC_CONFIG .....	208
Table 196: 0x35D AGC_MODE .....	208
Table 197: 0x35E AGC_LOW_THRESHOLD .....	209
Table 198: 0x35F AGC_HIGH_THRESHOLD .....	209
Table 199: 0x360 AGC_GAIN_STATUS .....	209
Table 200: 0x372 FREQUENCY_ERROR_READBACK .....	209
Table 201: 0x3CB VCO_BAND_OVRW_VAL .....	209
Table 202: 0x3CC VCO_AMPL_OVRW_VAL .....	209
Table 203: 0x3CD VCO_OVRW_EN .....	210
Table 204: 0x3D0 VCO_CAL_CFG .....	210
Table 205: 0x3D2 XOSC_CONFIG .....	210
Table 206: 0x3DA VCO_BAND_READBACK .....	210
Table 207: 0x3DB VCO_AMPL_READBACK .....	210
Table 208: 0x3F8 ANALOG_TEST_BUS .....	210
Table 209: 0x3F9 RSSI_TSTMUX_SEL .....	211
Table 210: 0x3FA GPIO_CONFIGURE .....	211
Table 211: 0x3FD TEST_DAC_GAIN .....	211
Table 212: GPIO Multiplex Table .....	213

Table 213: Internal Digital Signals .....	215
Table 214. Digital Test Mode Outputs.....	216
Table 215. Analog Test Mode Outputs .....	216
Table 216: GPIO Interface Memory Address Table .....	219
Table 217. Test Mode Access .....	221
Table 218: GPxCON Register Bit Description.....	221
Table 219: GPxOEN Register Bit Description .....	221
Table 220: GPxPUL Register Bit Description.....	222
Table 221: GPxOCE Register Bit Description .....	222
Table 222: GPxIN Register Bit Description .....	222
Table 223: GPxOUT Register Bit Description .....	223
Table 224: GPxSET Register Bit Description.....	223
Table 225: GPxCLR Register Bit Description.....	223
Table 226: GPxTGL Register Bit Description.....	224
Table 227. RFTST register bit description .....	224
Table 228: GPIO Port 1 Multiplex Table .....	225
Table 229: I <sup>2</sup> C Interface Memory Address Table Master Registers.....	231
Table 230: I <sup>2</sup> C Interface Memory Address Table Slave Registers.....	231
Table 231: I <sup>2</sup> C Interface Memory Address Table Shared Registers .....	231
Table 232: I2CMCON Register Bit Description .....	232
Table 233: I2CMSTA Register Bit Description .....	233
Table 234: I2CMRX Register Bit Description .....	234
Table 235: I2CMTX Register Bit Description.....	234
Table 236: I2CMRXCNT Register Bit Description .....	235
Table 237: I2CMCRXCNT Register Bit Description .....	235
Table 238: I2CADR1 Register Bit Description.....	235
Table 239: I2CADR2 Register Bit Description.....	236
Table 240: I2CSBYT Register Bit Description.....	236
Table 241: I2CDIV Register Bit Description .....	236
Table 242: I2CSCON Register Bit Description.....	237
Table 243: I2CSSTA Register Bit Description.....	239
Table 244: I2CSRX Register Bit Description.....	241
Table 245: I2CSTX Register Bit Description .....	241
Table 246: I2CALT Register Bit Description.....	241
Table 247: I2CIDx Register Bit Description.....	241
Table 248: I2CFSTA Register Bit Description .....	242
Table 249: I2CSHCON Register Bit Description .....	243
Table 250: SPIxCON[15:14] IRQ Mode Bits: .....	248
Table 251: SPI Peripheral Memory Address Table .....	251
Table 252: SPIxSTA Register Bit Description .....	252
Table 253: SPIxRX Register Bit Description .....	253

Table 254: SPIxTX Register Bit Description.....	253
Table 255: SPIxDIV Register Bit Description .....	254
Table 256: SPIxCON Register Bit Description .....	255
Table 257: SPIxDMA Register Bit Description .....	257
Table 258: SPIxCNT Register Bit Description.....	257
Table 259: UART Interface Memory Address Table.....	260
Table 260: COMRX/COMTX Register Bit Description.....	261
Table 261: COMIEN Register Bit Description .....	262
Table 262: COMIIR Register Bit Description.....	262
Table 263: Interrupt Identification Table.....	263
Table 264: COMLCR Register Bit Description .....	263
Table 265: COMMCR Register Bit Description .....	264
Table 266: COMLSR Register Bit Description .....	264
Table 267: COMMSR Register Bit Description .....	265
Table 268: COMFBR Register Bit Description .....	266
Table 269: Baudrate Examples .....	266
Table 270: COMDIV Register Bit Description .....	267
Table 271: COMCON Register Bit Description .....	267
Table 272: Timer Capture Event.....	270
Table 273: General Purpose Timer0 Memory Mapped Registers Address Table.....	271
Table 274: General Purpose Timer1 Memory Mapped Registers Address Table.....	271
Table 275: T0LD and T1LD Register Bit Description .....	271
Table 276: T0VAL and T1VAL Register Bit Description .....	272
Table 277: T0CON and T1CON Register Bit Description.....	272
Table 278: T0CLRI and T1CLRI Register Bit Description .....	273
Table 279: T0CAP and T1CAP Register Bit Description.....	273
Table 281: Wake Up Timer Memory Mapped Registers Address Table.....	277
Table 282: T2VAL Register Bit Description.....	277
Table 283: T2CON Register Bit Description.....	278
Table 284: T2INC Register Bit Description .....	279
Table 285: T2WUFB0 Register Bit Description .....	279
Table 286: T2WUFB1 Register Bit Description .....	279
Table 287: T2WUFC0 Register Bit Description.....	280
Table 288: T2WUFC1 Register Bit Description.....	280
Table 289: T2WUFD0 Register Bit Description.....	280
Table 290: T2WUFD1 Register Bit Description.....	280
Table 291: T2IEN Register Bit Description .....	281
Table 292: T2STA Register Bit Description .....	282
Table 293: T2CLRI Register Bit Description .....	283
Table 294: T2WUFA0 Register Bit Description .....	283
Table 295: T2WUFA1 Register Bit Description.....	283

Table 296: Watchdog Timer Memory Mapped Registers Address Table .....	285
Table 297: T3LD Register Bit Description .....	285
Table 298: T3VAL Register Bit Description .....	285
Table 299: T3CON Register Bit Description .....	286
Table 300: T3CLRI Register Bit Description .....	287
Table 301: T3STA Register Bit Description .....	287
Table 302: PWM Channel Grouping .....	288
Table 303: Compare Register Description .....	289
Table 304: PWM Equations .....	290
Table 305: PWM Memory Mapped Registers Address Table .....	291
Table 306: PWMCON0 Register Bit Description .....	292
Table 307: PWMCON1 Register Bit Description .....	293
Table 308: PWMCLRI Register Bit Description .....	293
Table 309: PWM Output in H-bridge Mode .....	294
Table 310: ADC Memory Mapped Registers Address Table .....	299
Table 311: ADCCFG Register Bit Description .....	300
Table 312: ADCCON Register Bit Description .....	301
Table 313: ADCSTA Register Bit Description .....	302
Table 314: ADCDAT Register Bit Description .....	302
Table 315: ADCGN Register Bit Description .....	302
Table 316: ADCOF Register Bit Description .....	302
Table 317: PSM Memory Mapped Registers Address Table .....	304
Table 318: PSMCON Register Bit Description .....	304
Table 319: ADuCRF101 Pin Function Descriptions .....	306
Table 320: SWD Connections .....	311
Table 321. Configuration of the External PA and LNA Control Signals .....	314

# List of Figures

Figure 1: ADUCRF101 High Level Block Diagram .....	20
Figure 2: Cortex-M3 Memory Map Diagram.....	21
Figure 4: Power Gating Diagram .....	33
Figure 6: Memory Map of Primary and Alternate DMA Structures.....	53
Figure 7: Information and User Space Memory Map on ADuCRF101.....	79
Figure 8: Uppermost Page of User Memory.....	79
Figure 9: UHF Transceiver Simplified Block Diagram .....	95
Figure 10: Radio Controller State Diagram for Normal Operation .....	98
Figure 11: Search for Preamble and Sync Word Routine.....	113
Figure 12: Transmit Sync Word Configuration .....	113
Figure 13: Payload Length in Fixed and Variable Length Packet Modes .....	116
Figure 14: Address Match Offset .....	116
Figure 15: Transmit Packet Timing .....	120
Figure 16: General SPORT Mode Packet.....	122
Figure 17: SPORT Mode Operation in Transmit Mode .....	122
Figure 18: SPORT Mode Receive, data_mode = 1 or 2.....	125
Figure 19: SPORT Mode Receive, data_mode = 1 .....	125
Figure 20: SPORT Mode Receive data_mode = 2.....	126
Figure 21: Overview of Interrupt Source and Mask Setup.....	127
Figure 22: UHF TRANSCEIVER Memory Map .....	131
Figure 23: Example Packet RAM Configurations: Tx and Rx Packet Address Pointers .....	133
Figure 24: SPI Inter Die Connections .....	134
Figure 25: Command Write (no parameters).....	135
Figure 27: Operation of the CMD_READY and FW_STATE Bits .....	138
Figure 28: Command Queuing and Operation of the CMD_READY and FW_STATE Bits..	138
Figure 29: SPI Memory Access Command/Address Format .....	139
Figure 30: Memory (MCR, BBRAM or Packet RAM) Block Write .....	140
Figure 31: Memory (MCR, BBRAM or Packet RAM) Random Address Write .....	140
Figure 32: Memory (MCR, BBRAM or Packet RAM) Block Read.....	141
Figure 33: Memory (MCR, BBRAM or Packet RAM) Random Address Read .....	141
Figure 34: Low Power Mode Operation .....	146
Figure 35: Low Power Mode Timing when using the WUC .....	149
Figure 37: Low Power Mode Timing when using the WUC, the Firmware Timer and SWM with Carrier Sense Only .....	150
Figure 38: Low Power Mode Timing when using the WUC, the Firmware Timer and SWM	150
Figure 39: Hardware Wake Up Controller (WUC) .....	151
Figure 40: Packet Structure with appended Reed-Solomon Error Check Code (ECC).....	155
Figure 41: ECB Mode .....	156
Figure 42: CBC Mode1 .....	156

Figure 43: Synthesizer Architecture .....	157
Figure 44: PA Ramp for Different PA_RAMP Settings .....	161
Figure 45: PA/LNA Interface with Single Ended PA Configuration .....	162
Figure 46: PA/LNA Interface with Differential PA Configuration .....	163
Figure 48: GPIO Structure .....	217
Figure 49: Slave read/write Flow Diagram .....	227
Figure 50: Slave Read/Write Flow Diagram Continued – Read Portion .....	228
Figure 51: SPI Transfer Protocol CPHA = 0.....	247
Figure 52: SPI Transfer Protocol CPHA = 1.....	247
Figure 53: Baudrate Generation .....	267
Figure 54: General Purpose Timers Block Diagram.....	269
Figure 55: Wake up Timer Block Diagram .....	275
Figure 56: Watchdog Timer Block Diagram .....	284
Figure 57: Waveform of PWM Channel Pair in Standard Mode .....	289
Figure 58: ADC Block Diagram.....	295
Figure 59: Equivalent Analog Input Circuit.....	296
Figure 60: ADC Timing Internal Signals.....	296
Figure 61: Typical Power-on Cycle.....	303
Figure 62: ADUCRF101 Pin Configuration .....	305
Figure 63: Simplified Supplies Connections.....	309
Figure 64: External Parallel Resonant Crystal Connections.....	310
Figure 65: SWD 20-PIN Connector Pinout.....	311
Figure 67: Combined differential PA and LNA match.....	314
Figure 71: Matching Topology for Transmit Antenna Diversity .....	314



# Introduction

## Purpose of Document

This User Guide provides reference information for the ADuCRF101 microcontroller, describing the functional blocks of the system in a package device designed around the ARM® Cortex™-M3 core.

This manual is intended for system software developers, hardware designers, and application developers. It is organized into sections that correspond to each major feature.

## Revision History

Revision	Release Date	Changes
PrA	05/11/2010	Initial Release

## Terminology

ADC	Analog to Digital Converter
AGC	Automatic Gain Control
AFC	Automatic Frequency Control
Battmon	Battery Monitor
BBRAM	Battery Back up Random Access Memory
CRC	Cyclic Redundancy Check
DR	Data Rate
FSK	Frequency Shift Keying
GFSK	Gaussian Frequency Shift Keying
MCR	Modem Configuration RAM
NOP	No Operation
NVIC	Nested Vectored Interrupt Controller
OOK	On-Off Keying
PA	Power Amplifier
PFD	Phase Frequency Detector
PHY	Physical Layer
PWM	Pulse Width Modulation
RCO	RC Oscillator
RISC	Reduced Instruction Set Computer
RSSI	Receive Signal Strength Indicator
Rx	Receive
SWD	Sync Word Detect/ Serial Wire Debug
SWM	Smart Wake Mode
Tx	Transmit
VCO	Voltage Controlled Oscillator
WUC	Wake Up Controller
XOSC	Crystal Oscillator

# Architectural Overview

## Product Features

The ADuCRF101 is a fully integrated System on Chip (SOC) solution designed for active low power wireless applications. It includes a 431-464MHz and 862-928MHz UHF transceiver, low power Cortex-M3 core from ARM and Flash/EE memory in a dual stacked die configuration, packaged in a 9mm x 9mm LFCSP.

The device can operate directly from a 3.6V battery and is specifically designed for low power operation.

The UHF transceiver is based on the ADF7023 device with single ended and differential input/output stages and communicating in the 431-464MHz and 862-928MHz frequency bands.

The ADuCRF101 integrates a low power Cortex-M3 core from ARM. It is a 32-bit RISC machine, offering up to 1.25 DMIPS peak performance. The Cortex-M3 MCU also has a flexible 14-channel DMA controller supporting all (SPI, UART, I<sup>2</sup>C) communication peripherals. 128k Bytes of non-volatile Flash/EE memory and 16k Bytes of SRAM are also provided on-chip.

The device operates from an on-chip oscillator generating an internal 16MHz high-frequency clock. This clock is routed through a programmable clock divider from which the MCU core clock operating frequency is generated.

The device also integrates a range of on-chip peripherals which can be configured under microcontroller software control as required in the application. These peripherals include UART, I2C and SPI Serial I/O communication controllers, 6-Channel ADC, GPIO Ports, 2 General Purpose Timers, Wake-up Timer and System Watchdog Timer and PWM.

The ADC consists of 6 single ended inputs used for ratio metric measurements on external sensors. These sensors can be powered temporarily for the measurements from the internal LDO. An internal battery monitor channel is also available.

The device is specifically designed to operate in battery powered applications where low power operation is critical. The device can be configured in normal operating mode or different low power modes under direct program control, including NONRETAINED mode (internal wake-up timer active). In NONRETAINED mode, external interrupts 0, 1, 2, and UHF transceiver interrupt can wake up the device. This allows the part to operate in an ultra-low power operating mode and still respond to active radio communication events.

On-chip factory firmware supports in-circuit serial download via the UART while non-intrusive emulation and program download is also supported via the serial wire interface.

These features are incorporated into a low-cost Development System supporting this Precision Analog Microcontroller family.

The parts operate from 1.8V to 3.6V and are specified over an industrial temperature range of -40°C to 85°C.

# ADuCRF101 Block Diagram

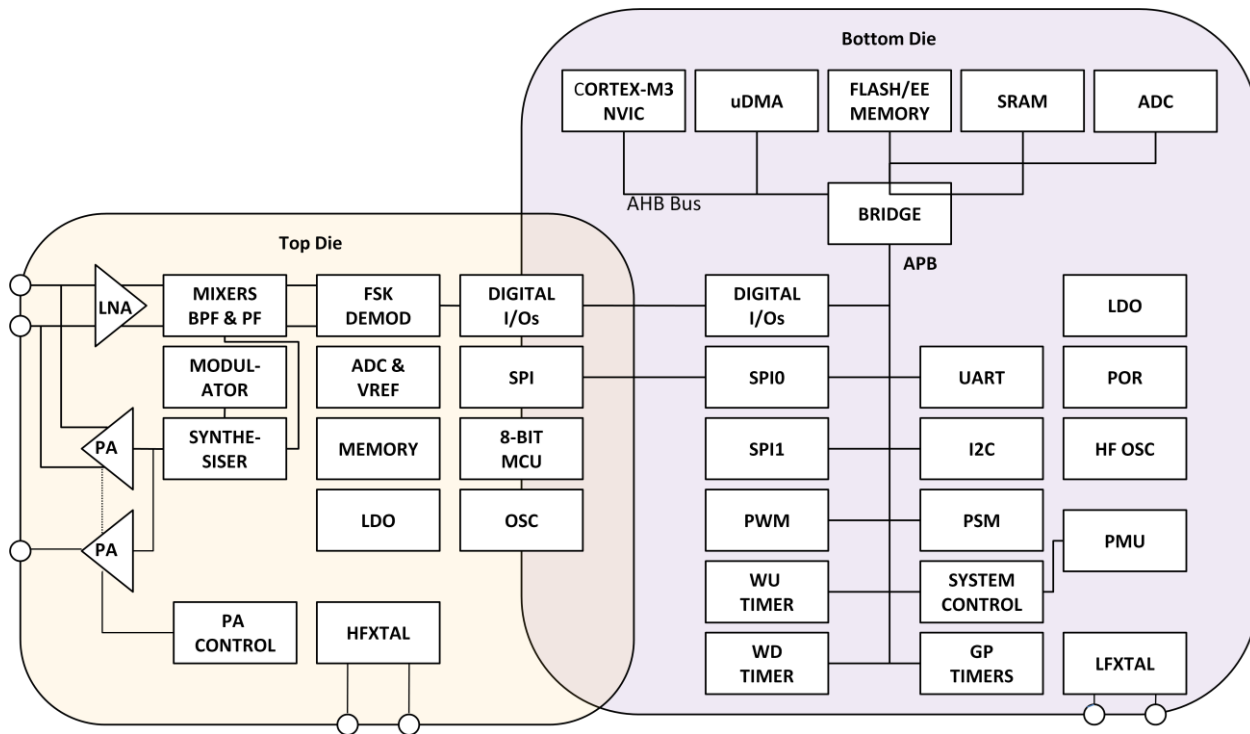


Figure 1: ADuCRF101 High Level Block Diagram

## Memory Organisation

The ADuCRF101 is a two die solution. The UHF transceiver on the top die has its own memory organisation. The bottom die (the cortex die) memory organisation is described in this section.

Three separate blocks of memory are accessible to the user:

1. 16kB of SRAM from 0x20000000 to 0x20003FFF
2. 128kB of on-chip Flash/EE memory available to the user from 0 to 0x1FFFF
3. An additional 2kB reserved for the kernel space from 0x20000 to 0x207FF.

These blocks are mapped as per cortex memory map as shown in Figure 2 . All on-chip peripherals are accessed via memory mapped registers, situated in the bit band region and described later in this document.

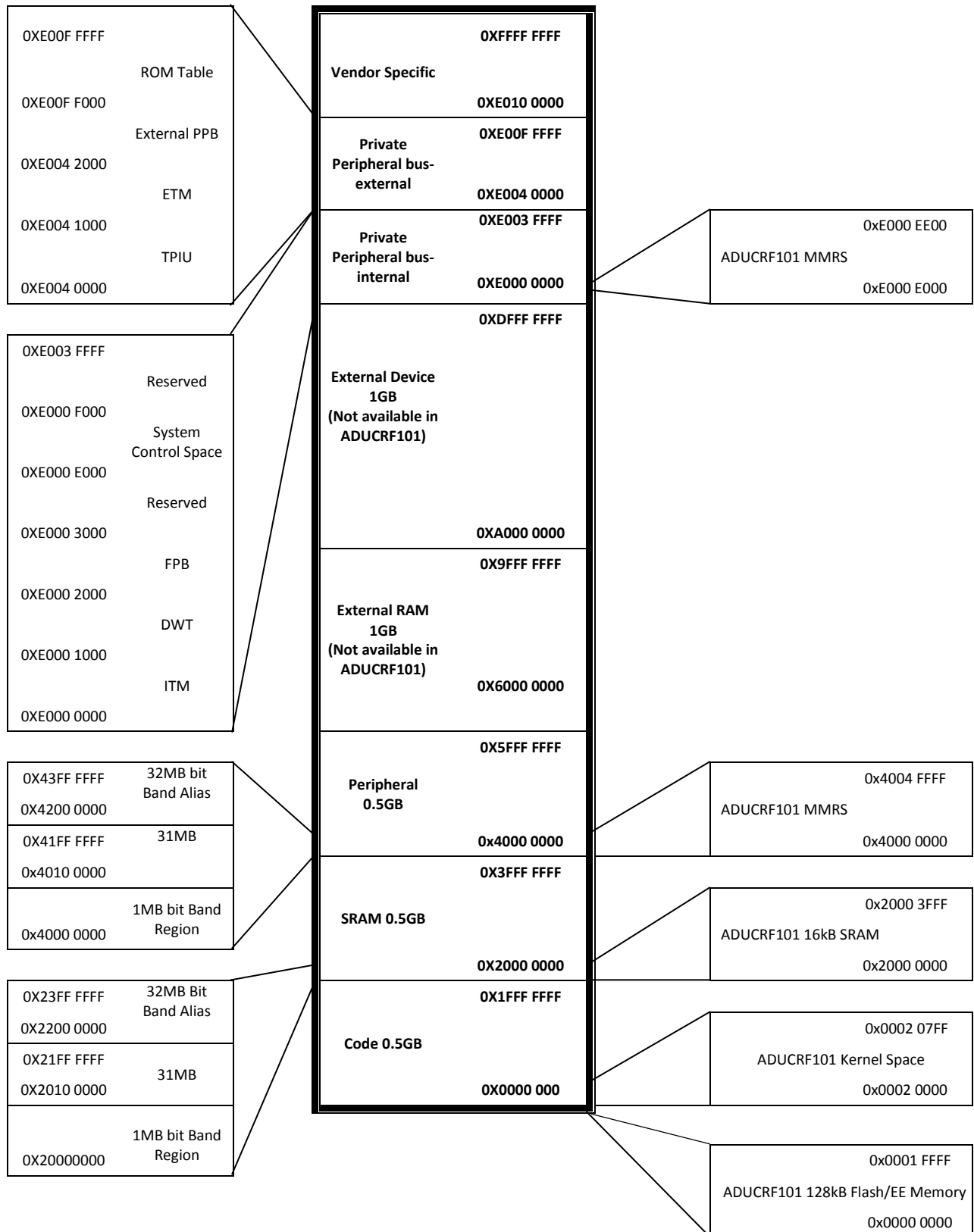


Figure 2: Cortex-M3 Memory Map Diagram

**ADuCRF101 Memory Map Register Summary****Table 1: ADuCRF101 Memory Map Register Summary**

First Address	Last Address	Peripheral
0x40000000	0x4000001C	General Purpose Timer0
0x40000400	0x4000041C	General Purpose Timer1
0x40001000	0x4000104C	PWM
0x40002000	0x40002028	System Control
0x40002400	0x40002410	Always On Section: Power Control
0x40002420	0x40002434	Always On Section: External Interrupts
0x40002440	0x40002440	Always On Section: Reset
0x40002500	0x40002540	Always On Section: Wake up Timer
0x40002580	0x40002598	Always On Section: Watchdog Timer
0x40002800	0x4000287C	Flash Controller
0x40003000	0x40003050	I2C
0x40004000	0x40004018	SPI0
0x40004400	0x40004418	SPI1
0x40005000	0x40005030	UART
0x40006000	0x400060E4	Digital I/Os
0x40009C00	0x40009C20	HF Oscillator Trim
0x40010000	0x40010FFC	UDMA
0x40050000	0x40050014	ADC Interface
0xE000E100	0xE000E014	NVIC

# System Clocks

## System Clock Features

The ADuCRF101 integrates 2 on-chip oscillators and circuitry for 2 external crystals:

- HFXTAL is a 26 MHz external crystal, used for the UHF transceiver.
- LFOSC is a 32 kHz low power internal oscillator, used in low power modes.
- HFOSC is a 16 MHz internal oscillator, used in active mode.
- LFXTAL is a 32 kHz external crystal, clocking the wake up timer. This circuitry can also support a 65 kHz crystal.

A trimming feature is available to adjust the 16 MHz oscillator frequency during code execution.

Three of the clock sources, LFOSC, HFOSC and LFXTAL, can be used as system clocks. An external clock on P0.5 can also be used for test purposes.

Internally system clock is divided into 5 clocks:

- FCLK for the core
- UCLK system clock
- HCLK for the AHB peripherals (Flash, SRAM, DMA)
- ACLK for the ADC interface
- PCLK for the APB peripherals.

Figure 3 shows all the clocks available and includes clock gates for power management. More details on the clock gates are in the power management section.

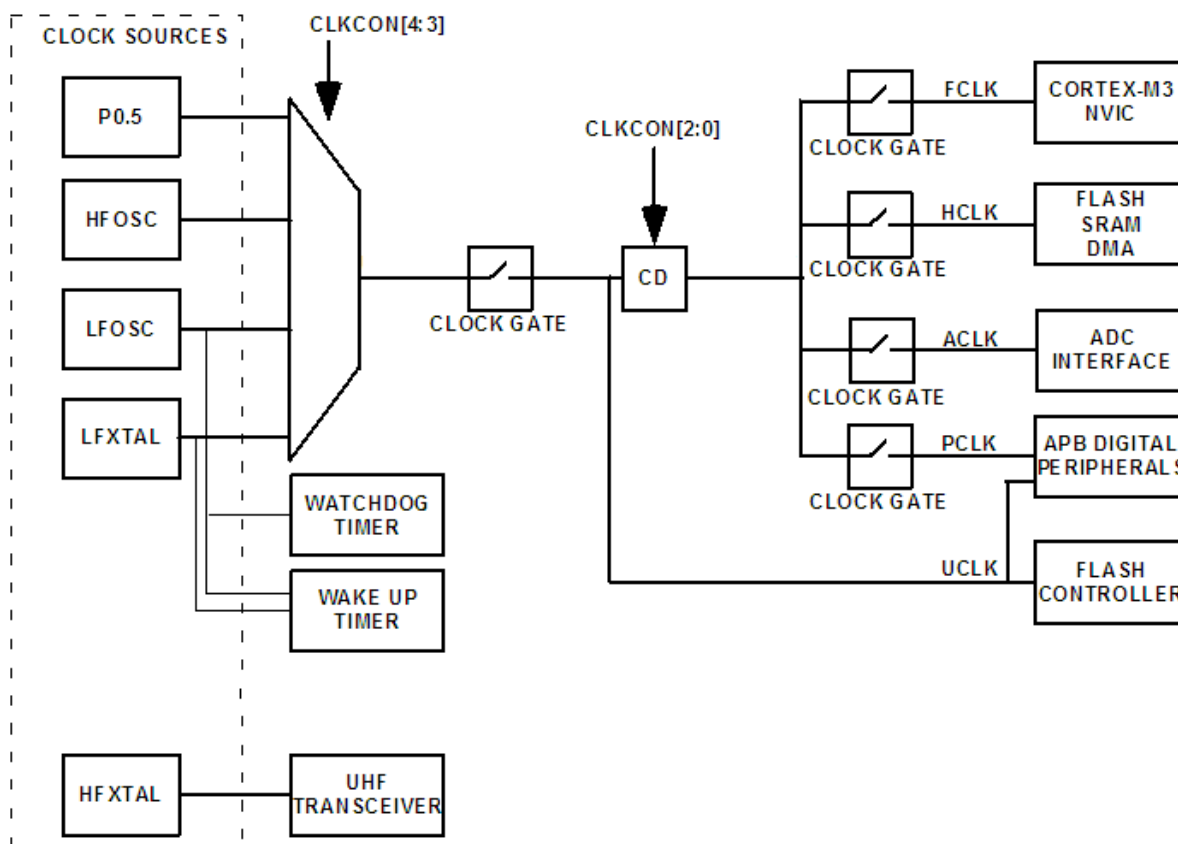


Figure 3: System Clock Architecture Block Diagram

## System Clocks Operation

At power up the core executes from the 16MHz internal oscillator. User code can select the clock source for the system clock and can divide the clock by a factor of  $2^{CD}$ . Where CD is the clock divider bits (CLKCON [2:0]). This allows slower code execution and reduced power consumption.

UCLK is also passed to some of the serial peripherals so that the timings are not affected by CD changes.

The two internal oscillators, external crystal, clocks from the top die or UCLK can be brought out to P0.4 for test purposes.

The external crystal circuitry can also support a 65 kHz crystal. This clock source can be divided by two if necessary in the XOSCCON register.

Note that P0.5 must be configured first as a clock input before switching clock source in the clock control register.



## 16MHz Oscillator Trim

The internal 16MHz oscillator can drift slightly over temperature. The temperature drift can be measured and compensated for using the 32 kHz crystal oscillator as a reference.

The trim is carried out using two timers. One timer will count a number of crystal clocks to generate an accurate timebase. The second timer will simultaneously count the number of 16MHz clocks during this timebase. Once the timebase has ended, the count in the second timer will be compared with a reference count number.

If the count is longer than the reference, the oscillator trim value will be incremented.

If the count is shorter than the reference, the oscillator trim value will be decremented.

There are a number of interrupt sources to detect various conditions:

- Crystal timer end ( Enabled in HFTCON[0])
- 16 MHz timer overflow( Enabled in HFTCON[1])
- Maximum trim code reached ( Enabled in HFTCON[2])
- Minimum trim code reached ( Enabled in HFTCON[3])

In normal operation it is not necessary to service the ISR. The interrupt source is provided for debug and monitoring purposes.

The maximum trim code is set at final test by ADI. This is to prevent over clocking of the device. The user cannot modify this value.

User code cannot write a trim code that exceeds the maximum limit. If this is attempted the register will receive the max trim value instead.

The min trim code will be user programmable.

On entering a power mode which disables the 16MHz oscillator the current trim cycle is aborted and both counters are reset to zero but the trim value is unaffected.

At the end of the time base period the contents of HFTUMAX and HFTUVAL is compared with the tolerance specified by TOL.

IF  $HFTUVAL > HFTUMAX + 32 \times (1 + TOL)$  the decrement the trim value by 1

IF  $HFTUVAL < HFTUMAX - 32 \times (1 + TOL)$  the increment the trim value by 1

For example:

(The values below are purely for calculation purposes and are not representative of real applications)

UCLK count = 900

HFTUMAX = 1000

HFTXMAX = 2

For a tolerance of 32 counts TOL = 0 (HFTCON [5] = 0)

$HFTUMAX + 32 \times (1 + TOL) = 1000 + 32 \times (1 + 0) = 1000 + (32 \times 1) = 1032$

$HFTUMAX - 32 \times (1 + TOL) = 1000 - 32 \times (1 + 0) = 1000 - (32 \times 1) = 968$

If UCLK count = 900 then

$900 > 1032 = \text{False}$

$900 < 968 = \text{True}$ , therefore increment the trim value by 1

If UCLK count = 999 then

$999 > 1032 = \text{False}$

$999 < 968 = \text{False}$

Then the trim register does not have to be incremented or decremented

If UCLK count = 1050 then

$1500 > 1032 = \text{True}$ , therefore decrement the trim value by 1

$1500 < 968 = \text{False}$

For a tolerance of 64 counts  $TOL = 1$  (HFTCON [5] = 0)

$HFTUMAX + 32 \times (1 + TOL) = 1000 + 32 \times (1 + 1) = 1000 + (32 \times 2) = 1064$

$HFTUMAX - 32 \times (1 + TOL) = 1000 - 32 \times (1 + 1) = 1000 - (32 \times 2) = 936$

The value of HFTUMAX can be calculated as follows:

$HFTUMAX = HFTXMAX \times F_{UCLK}/F_{XTAL}$

$HFTUMAX = 2 \times (16M/32K)$

$HFTUMAX = 1000$

# System Clocks Memory Mapped Registers

The system clocks are controlled by registers based at different addresses.

**Table 2: Clock Control Memory Mapped Registers Address Table**

Address	Name	Description	Access	Default
0x40002000	CLKCON	System clock control register	RW	0x0000
0x40002410	XOSCCON	Crystal oscillator control register	RW	0x00

## System Clocks Control Register: CLKCON

**Table 3: CLKCON Register Bit Description**

Address: 0x40002000

Bits	Name	Description
15 to 8	Reserved	Reserved
7 to 5	CLKOUT	Clock out multiplexer selection bits 000: UCLKCG (default) 001: UCLK 010: PCLK 011: Reserved 100: Reserved 101: HFOSC 110: LFOSC 111: LFX TAL
4 to 3	CLKMUX	Clock in multiplexer selection bits 00: HFOSC (default) 01: LFX TAL 10: LFOSC 11: EXTP05
2 to 0	CD	Clock divide bits: 000: /1 = 16MHz(default) 001: /2 = 8MHz 010: /4 = 4MHz 011: /8 = 2MHz 100: /16 = 1MHz 101: /32 = 500kHz 110: /64 = 250 kHz 111: /128 = 125kHz

## External Crystal Oscillator Control Register: XOSCCON

**Table 4: XOSCCON Register Bit Description**

Address: 0x40002410

Bits	Name	Description
7 to 3	Reserved	These bits are Reserved and should be written 0.
2	DIV2	Divide by two enable bit. 1: Enable the clock divider. 0: Disable the clock divider.
1	Reserved	This bit is Reserved and should be written 0.
0	ENABLE	Crystal oscillator circuit enable 1: Enable the oscillator circuitry. 0: Disable the oscillator circuitry.

## HF Oscillator Trim Registers

**Table 5: HF Oscillator Trim Registers Address Table**

Base Address: 0x40009C00

Offset	Name	Description	Access	Default
0x00	HFTSTA	Status register	R	0x00
0x04	HFTCON	Control Register	RW	0x00
0x08	HFTMAX	Maximum calibration value	R	-
0x0C	HFTMIN	Minimum calibration value	RW	0x00
0x10	HFTTRM	Oscillator trim value	RW	MAXCAL
0x14	HFTXMAX	Xtal target count	RW	0x0
0x18	HFTXVAL	XTAL current count	R	0x0
0x1C	HFTUMAX	UCLK target count	RW	0x000
0x20	HFTUVAL	UCLK current count	R	0x000

**HFOSC Trim Status Register: HFTSTA**

**Table 6: HFTSTA Register Bit Description**  
Address 0x40009C00

Bits	Name	Description
5	DEC	The last calibration cycle was a decrement. (even when minimum reached or STICK set)
4	INC	The last calibration cycle was an increment. (even when maximum reached or STICK set)
3	MIN	The minimum calibration trim value has been reached. (HFT Interrupt source)
2	MAX	The maximum calibration trim value has been reached. (HFT Interrupt source)
1	OF	UCLK count overflow (HFT Interrupt source)
0	END	Calibration cycle ended (HFT Interrupt source)

**HFOSC Trim Control Register: HFTCON**

**Table 7: HFTCON Register Bit Description**  
Address 0x40009C04

Bits	Name	Description
7	CLR	1: the ENABLE bit will be automatically cleared at the end of the next calibration cycle.
6	STICK	1: the trim register will not be incremented or decremented at the end of a calibration cycle.
5	TOL	Select the compare tolerance at the end of the count cycle. 1: use a tolerance of 32 count 0: use a tolerance of 64 counts If the UCLK counter and the user supplied expected value agree within this tolerance then no increment or decrement will occur.
4	ENABLE	1: Enable this calibration block( Trim).
3	MINIEN	1: Enable the minimum trim value interrupt.
2	MAXIEN	1: Enable the maximum trim value interrupt.
1	OFIEN	1: Enable the UCLK counter overflow interrupt.
0	ENDIEN	1: Enable the cycle end interrupt.

Note: If the trim is enabled (HFTCON [4] = 1) and CLR (HFTCON [7]) is not set the part will keep calibrating. That is adjusting the trim value and generating cycle end interrupts.

**Maximum Calibration Value: HFTMAX**

**Table 8: HFTMAX Register Bit Description**  
Address 0x40009C08

Bits	Name	Description
7 to 0	MAX	Maximum trim value The value in this register is programmed in ADI. It is not user programmable The trim routine will not increment past this value

**Minimum Calibration Value: HFTMIN**

**Table 9: HFTMIN Register Bit Description**  
Address 0x40009C0C

Bits	Name	Description
7 to 0	MIN	Minimum trim value. The trim routine will not increment past this value

**Oscillator Trim Value: HFTTRM**

**Table 10: HFTTRM Register Bit Description**  
Address 0x40009C10

Bits	Name	Description
7 to 0	TRM	Trim value. The value written to this register can not exceed MAX. If a value of greater than MAXCAL is written then the MAX value is applied. Is is possible to write a value that is less than MINCAL. The register is writable when the trim is not enabled The register is read-only when the trim is enabled and not cleared. The register will increase or decrease by 1 as necessary for every calibration cycle.

**XTAL Target Count: HFTXMAX**

**Table 11: HFTXMAX Register Bit Description**  
Address 0x40009C14

Bits	Name	Description
7 to 0	XMAX	The number of crystal clocks to count. Cleared means disabled

**XTAL Current Count: HFTXVAL****Table 12: HFTXVAL Register Bit Description**

Address 0x40009C18

Bits	Name	Description
7 to 0	XVAL	The current count of the crystal clocks. Realtime updating

**UCLK Target Count: HFTUMAX****Table 13: HFTUMAX Register Bit Description**

Address 0x40009C1C

Bits	Name	Description
12 to 0	UMAX	The expected number of 16 MHz clocks during the timebase

**UCLK Current Count: HFTUVAL****Table 14: HFTUVAL Register Bit Description**

Address 0x40009C20

Bits	Name	Description
12 to 0	UVAL	The current count of 16MHz clocks during the current timebase. It holds the count when each calibration cycle finishes

# Power Management Unit

## Power Management Unit Features

The Power Management Unit (PMU) controls the power modes of the ADuCRF101 with the exception of UHF transceiver. The UHF transceiver power modes are independent and are described in the UHF transceiver section. The Cortex-M3 sleep modes are linked to the PMU modes and are described in this section.

Two techniques are used to reduce power to a minimum on the ADuCRF101: clock gating and power gating. A clock gating diagram is shown in the system clock section (Figure 3). A power gated diagram is shown in Figure 4. The PMU is in the “Always On” section. 7 power modes are available. Each mode gives a power reduction benefit with a corresponding reduction in functionality.

Current values are available in the datasheet.

The Cortex-M3 has two power saving modes:

- Sleep mode: stops the system clock (FCLK)
- Deep sleep mode: stops the system clock (FCLK) and in conjunction with the PMU, switches off some circuitry like UCLK, HFOSC and the HP LDO.

The WFI instruction places the cortex in sleep mode or deep sleep mode depending on the sleepdeep bit in the cortex M3 system control register (0xE000ED10).

If deep sleep mode is enabled in the system control register of the cortex when issuing the WFI instruction, the Cortex will enter deep sleep mode, else it will be in sleep mode.

Enabling deep sleep mode in mode 1, 2 or 3 does not have any effect. Not enabling deep sleep mode in mode 4, 5 or 6 will leave the ADuCRF101 in active mode.

The PMU and Cortex-M3 modes are summarized in Table 15.



## Power Management Unit Block Diagram

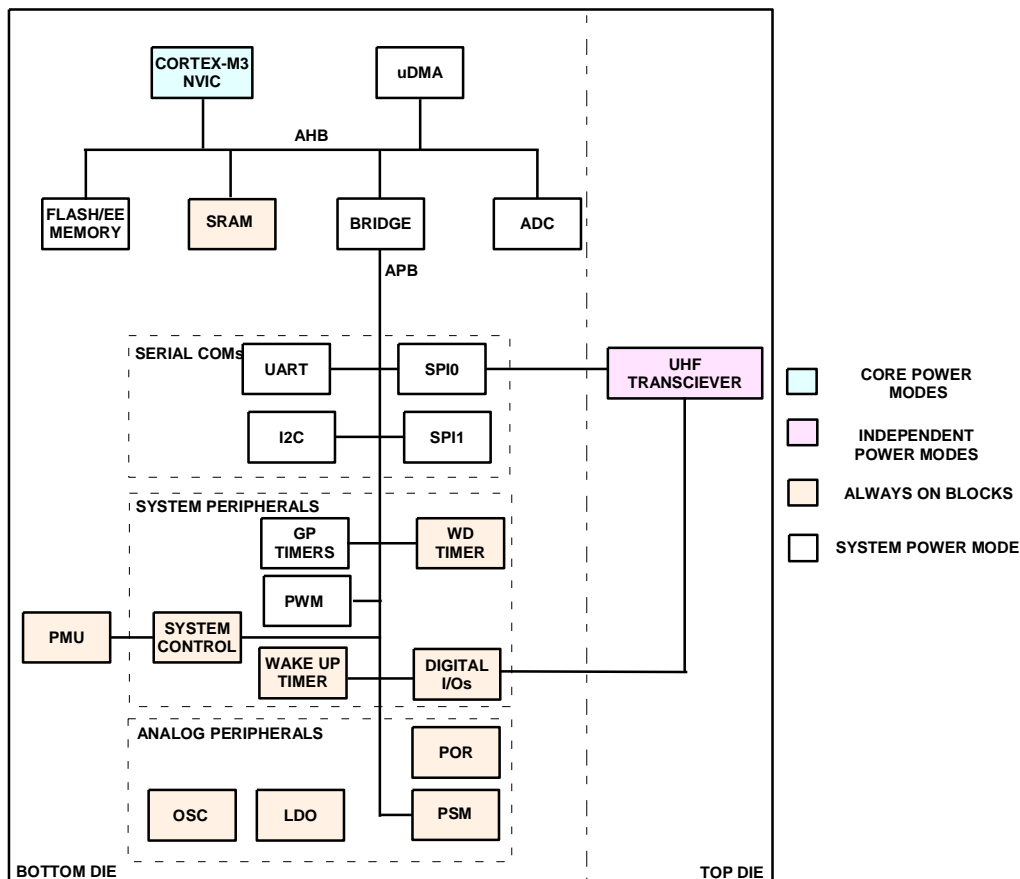


Figure 4: Power Gating Diagram

All communication peripherals should be disabled before entering low power modes where PCLK is disabled. Disabling the peripheral will reset the state machine of these peripheral while keeping their configuration. This ensures that on wake up the communication peripherals are in a known state.

## Power Management Unit LDO switching

Two LDO regulators are integrated on the ADuCRF101.

- The high power LDO (HP LDO) will supply the part in its active mode.
- The low power LDO (LP LDO) achieves an extremely low quiescent current, however it can only supply very low load currents (<50μA). It is used in modes 4 and 5.

Switching between the LDOs is done automatically and is transparent to the user.

## Power Management Unit Power Modes Operation

### Power Mode: ACTIVE mode

The system is fully active. None of the 5 clocks described in Figure 3 are gated. Memories and all user enabled peripherals are clocked and the Cortex-M3 is executing instructions. Note that the Cortex-M3 manages its internal clocks and can be in a partial clock gated state. This clock gating only effects the internal Cortex-M3 processing core. Automatic clock gating is used on all blocks and is transparent to the user. User code can use a WFI command to sleep the Cortex-M3; it is independent of the power mode settings of the PMU.

When the ADuCRF101 wakes up from any of the low power mode, it returns to ACTIVE mode.

### Power Mode: CORTEX mode, mode 0

None of the 5 clocks described in Figure 3 are gated. Memories and all user enabled peripherals are clocked. The Cortex-M3 does not execute instructions. Executing a WFI instruction to power mode 0 places the cortex in this sleep mode.

### Power Mode: MCUHALT mode, mode 1

The system gates HCLK at an early stage after the Cortex-M3 has entered SLEEP mode. The HCLK is the AMBA AHB, “Advanced High-speed Bus” clock. The gating of this clock stops all AHB attached masters and slaves. The Cortex-M3 FCLK is active and the device will wake up using the NVIC.

### Power Mode: PERHALT mode, mode 2

The system gates PCLK at an early stage after the Cortex-M3 has entered SLEEP mode. The PCLK is the AMBA APB, “Advanced Peripheral Bus” clock. The gating of this clock stops all APB attached slave peripherals. The Cortex-M3 FCLK is active and the device will wake up using the NVIC.

### Power Mode: SYSHALT mode, mode 3

The system gates HCLK and PCLK at an early stage after the Cortex-M3 has entered SLEEP mode. The HCLK is the AMBA AHB, “Advanced High-speed Bus” clock. The PCLK is the AMBA APB, “Advanced Peripheral Bus” clock. The gating of these clocks stops all AHB attached masters/slaves and all APB attached slave peripherals. The Cortex-M3 FCLK is active and the device will wake up using the NVIC.

### Power Mode: TOTALHALT mode, mode 4

The system gates FCLK, HCLK and PCLK at an early stage after the Cortex-M3 has entered SLEEPDEEP mode. The FCLK is the Cortex-M3 free-running clock. The HCLK is the AMBA AHB, “Advanced High-speed Bus” clock. The PCLK is the AMBA APB, “Advanced Peripheral Bus” clock. The gating of these clocks stops some of the NVIC functionality, all AHB attached masters/slaves, and all APB attached slave peripherals. The Cortex-M3 FCLK is thus stopped and only the peripherals listed in Table 21 will be able to wake up the Cortex-M3. The 16MHz oscillator is also stopped in this mode. The low power LDO is turned on and the high power LDO turned off automatically when entering mode 4.

**Power Mode: HIBERNATE mode, mode 5**

The system gates power to the digital core. All states are retained during this power gating. It will appear to the user as a clock gating of FCLK, HCLK and PCLK at an early stage after the Cortex-M3 has entered SLEEPDEEP mode but with a lower leakage current. There will be a response time difference; the device will be slower to come up. The Cortex-M3 FCLK is thus stopped and only the peripherals listed in Table 21 will be able to wake up the Cortex-M3.

**Power Mode: NONRETAINED mode, mode 6**

In NONRETAINED mode none of the states are retained. All digital logic is off. Only POR and LPOSC are on. External interrupts can wake up the device. After waking up from mode 6 the on-chip firmware executes and all registers are in default values.

**Table 15: Power Modes Summary**

Clock and Power	Active	Mode 0	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
HP LDO	ON	ON	ON	ON	ON	OFF	OFF	OFF
LP LDO	OFF	OFF	OFF	OFF	OFF	ON	ON	OFF
HFOSC	ON	ON	ON	ON	ON	OFF	OFF	OFF
LFOSC	ON	ON	ON	ON	ON	ON	ON	ON
POWER GATE	ON	ON	ON	ON	ON	ON	OFF	OFF
UCLK	ON	ON	ON	ON	ON	OFF	OFF	OFF
FCLK	ON	ON	ON	ON	ON	OFF	OFF	OFF
HCLK	ON	ON	OFF	ON	OFF	OFF	OFF	OFF
PCLK	ON	ON	ON	OFF	OFF	OFF	OFF	OFF
ACLK	ON	ON	ON	ON	ON	OFF	OFF	OFF
SRAM	ON	ON	ON	ON	ON	ON	ON	OFF
CORTEX M3	Active	Sleeping	Sleeping	Sleeping	Sleeping	Sleepdeep	Sleepdeep	Sleepdeep

**Table 16: System Power Mode Summary**

Peripheral	HCLK	PCLK
uDMA	✓	
ADC	✓	
UART		✓
I <sup>2</sup> C		✓
SPI		✓

## Power Modes Memory Mapped Registers

The power modes are controlled by a single register based in the always on section at address 0x40002400.

**Table 17: System Clocks Memory Mapped Register Address**

Base Address: 0x40002400

Offset	Name	Description	Access	Default
0x0000	PWRMOD	Power modes register	RW	0x40
0x0004	PWRKEY	Key protection for PWRMOD	RW	N/A

### Power modes control register: PWRMOD

**Table 18: PWRMOD Register Bit Description**

Address: 0x40002400

Bits	Name	Description
7 to 3	Reserved	Reserved. These bits should be written 0 by user code.
2 to 0	MOD	Power modes control bits: Selects the power mode we want to enter. 000 = CORTEX 001 = MCUHALT 010 = PERHALT 011 = SYSHALT 100 = TOTALHALT 101 = HIBERNATE 110 = NONRETAINED Other = Reserved

To place the cortex in “sleepdeep” mode for modes 4, 5 and 6 the cortex M3 system control register (address 0xE000ED10) must be configured to 1 in bit 2.

### Power modes key register: PWRKEY

**Table 19: PWRKEY Register Bit Description**

Address: 0x40002400

Bits	Name	Description
15 to 0	VALUE	Power control key register. The PWRMOD register is key-protected. Two writes to the key are necessary to change the value in the PWRMOD register. First 0x4859, then 0xF27B. The PWRMOD register should then be written. A write to any other register on the APB bus (see Figure 1 before writing to PWRMOD) will return the protection to the lock state.

# Cortex-M3 Core

## Cortex-M3 Core Features

The ADuCRF101 contains an embedded ARM® Cortex™-M3 processor. It has been developed to provide a low-cost platform that meets the needs of MCU implementation, with a reduced pin count and low-power consumption, while delivering outstanding computational performance and an advanced system response to interrupts.

The ARM® Cortex™-M3 32-bit RISC processor features exceptional code-efficiency, delivering the high-performance expected from an ARM® core in the memory size usually associated with 8- and 16-bit devices.

### High Performance

- 1.25 DMIPS/MHz
- Many instructions, including multiply are single cycle
- Separate data and instruction busses allow simultaneous data and instruction accesses to be performed
- Optimized for single cycle flash usage

### Low Power

- Low standby current. Part includes a low power wake-up timer.
- Core implemented using advanced clock gating so only actively used logic consumes dynamic power.
- Power-saving mode support (SLEEPING and SLEEPDEEP). The design has separated clocks to allow unused parts of the core to be stopped.

### Advanced Interrupt-Handling

- The Nested Vectored Interrupt Controller (NVIC) supports up to 240 interrupts. The ADuCRF101 supports 43 of these. The vectored interrupt feature greatly reduces interrupt latency because there is no need for software to determine which interrupt handler to serve. In addition, there is no need to have software to set up nested interrupt support.
- The Cortex-M3 processor automatically pushes registers onto the stack at interrupt entry and pops them back at interrupt exit. This reduces interrupt handling latency and allows interrupt handlers to be normal C functions.
- Dynamic priority control for each interrupt.
- Latency reduction using late arrival interrupt acceptance and tail-chain interrupt entry.
- Immediate execution of a Non-maskable interrupt request for safety-critical applications.

### System Features

- Support for bit-band operation and unaligned data access.
- Advanced fault handling features include various exception types and fault status registers.

### **Debug Support**

- Serial Wire debug interfaces (SW-DP).
- Flash Patch and Breakpoint (FPB) unit for implementing breakpoints.
- Data Watchpoint and Trigger (DWT) unit for implementing watchpoints, trigger resources and system profiling.

# Cortex-M3 Core Functional Description

The ARM® Cortex™-M3 Technical Reference Manual describes all the features of an ARM Cortex-M3 in detail. However, these features differ based on the implementation. This section describes the implementation on the ADuCRF101. As noted in the *ARM® Cortex™-M3 Technical Reference Manual*, several Cortex-M3 components are flexible in their implementation: SW/JTAG-DP, ETM, TPIU, the ROM table, the MPU, and the Nested Vectored Interrupt Controller (NVIC). The following sections detail the actual implementation of these in the ADuCRF101.

## Serial Wire Debug (SW/JTAG-DP)

The part only supports the Serial Wire interface via the SWCLK and SWDIO pins. It does not support the 5 wire JTAG interface. See the *CoreSight™ Design Kit Technical Reference Manual* for details on SWJ-DP.

The Serial Wire interface is available on the two SWD dedicated pins only.

## ROM Table

The default ROM table was implemented as described in the *ARM® Cortex™-M3 Technical Reference Manual*.

## Nested Vectored Interrupt Controller Interrupts (NVIC)

The Cortex™-M3 processor includes an interrupt controller called the Nested Vectored Interrupt Controller (NVIC). It is closely coupled with the processor core and provides a number of features:

- Nested interrupt support
- Vectored interrupt support
- Dynamic priority changes support
- Interrupt masking

In addition, the NVIC also has a Non-maskable Interrupt (NMI) input. The NVIC is implemented on the ADuCRF101 and is described in more detail in the interrupt section.

## Wake-up Interrupt Controller (WIC)

ADI has implemented a modified WIC which provides the lowest possible power-down current.

## μDMA

The ADuCRF101 implements the ARM® μDMA. More details are in the DMA section.

For more information on the ARM Cortex-M3 processor core, see the ARM® Cortex™-M3 Technical Reference Manual (DDI0337G\_cortex\_m3\_r2p0\_trm.pdf) and the ARM® Cortex™-M3 User Guide Reference Material (DUI0450A\_ref\_material\_for\_cm3\_ug.pdf)

# Exceptions & Interrupts

## Cortex-M3 and Fault Management

The Cortex-M3 supports a number of system exceptions and peripherals/external interrupts as shown in Table 20 and in Table 21.

**Table 20: List of System Exceptions**

Number	Type	Priority	Description
1	Reset	-3 (highest)	Any reset
2	NMI	-2	Non-maskable interrupt connected to Power Supply Monitor of ADuCRF101.
3	Hard Fault	-1	All fault conditions, if the corresponding fault handler is not enabled.
4	Memory Management Fault	Programmable	Memory management fault; access to illegal locations.
5	Bus Fault	Programmable	Pre-fetch fault, memory access fault, and other address/memory related faults.
6	Usage Fault	Programmable	Faults such as undefined instruction executed or illegal state transition attempt.
7 to 10	Reserved	NA	
11	SVCall	Programmable	System service call with SVC instruction.
12	Debug Monitor	Programmable	Debug monitor (breakpoint, watchpoint, or external debug requests)
13	Reserved	NA	
14	PendSV	Programmable	Pendable request for system service.
15	SYSTICK	Programmable	System tick timer.



## Nested Vectored Interrupt Controller

ADuCRF101 interrupts are controlled by the NVIC, and 8 levels of priority are available. Only a limited number of interrupt can wake up the MCU from low power modes 4, 5 and 6. These are described in Table 21. When the device is woken up from any mode it returns to active mode.

**Table 21: Interrupt Vector Table**

Position number	Vector	Wake up MCU from mode 4/5	Wake up MCU from mode 6
0	Wake Up Timer	Yes	No
1	External interrupt 0	Yes	Yes
2	External interrupt 1	Yes	Yes
3	External interrupt 2	Yes	No
4	External interrupt 3	Yes	No
5	External interrupt 4	Yes	No
6	External interrupt 5	Yes	No
7	External interrupt 6	Yes	No
8	External interrupt 7	Yes	No
9	External interrupt 8	Yes	No
10	Watchdog timer	Yes	No
11	Reserved	-	-
12	Timer0	No	No
13	Timer1	No	No
14	ADC	No	No
15	Flash Controller	No	No
16	UART	No	No
17	SPI0, interface to UHF transceiver	No	No
18	SPI1, user SPI	No	No
19	I2C slave	No	No
20	I2C master	No	No
21	Reserved	-	-
22	DMA error	No	No
23	DMA SPI1 TX IRQ	No	No
24	DMA SPI1 RX IRQ	No	No
25	DMA UART TX IRQ	No	No

Interrupt vector table continued

Position number	Vector	Wake up MCU from mode 4/5	Wake up MCU from mode 6
26	DMA UART RX IRQ	No	No
27	DMA I2C slave TX	No	No
28	DMA I2C slave RX	No	No
29	DMA I2C master TX	No	No
30	DMA I2C master RX	No	No
31 to 33	Reserved	-	-
34	ADC DMA	No	No
35	DMA SPI0 TX IRQ	No	No
36	DMA SPI0 RX IRQ	No	No
37	PWM TRIP	No	No
38	PWM0	No	No
39	PWM1	No	No
40	PWM2	No	No
41	PWM3	No	No
42	HFT Interrupt	No	No

Note: The HFT interrupt is generated when the calibration cycle ends or the counter overflowed or maximum/minimum trim value reached. The interrupts are enabled in HFTCON [3:0].

Internally, the highest user-programmable priority (0) is treated as fourth priority, after a Reset, NMI, and a Hard Fault. Note that 0 is the default priority for all the programmable priorities.

If the same priority level is assigned to two or more interrupts, their hardware priority (the lower the position number) determines the order in which the processor activates them. For example, if both SPI0 and SPI1 are priority level 1, then SPI0 has higher priority.

Table 22 lists the useful registers to enable and disable relevant interrupts and set the priority levels:

Table 22: NVIC Registers

Address	ADI header file name	Description	Access
0xE000E100	ISER0	Set IRQ 0 to 31 enable	Read/write
0xE000E104	ISER1	Set IRQ 32 to 42 enable	Read/write
0xE000E180	ICER0	Clear IRQ 0 to 31 enable	Read/write
0xE000E184	ICER1	Clear IRQ 32 to 42 enable	Read/write
0xE000E200	INTSETP0	Set IRQ 0 to 31 pending	Read/write
0xE000E204	INTSETP1	Set IRQ 32 to 42 pending	Read/write
0xE000E280	ICPR0	Clear IRQ 0 to 31 pending	Read/write
0xE000E284	ICPR1	Clear IRQ 32 to 42 pending	Read/write
0xE000E400	IPR0	IRQ 0 to 3 Priority	Read/write
0xE000E404	IPR1	IRQ 4 to 7 Priority	Read/write
0xE000E408	IPR2	IRQ 8 to 11 Priority	Read/write
0xE000E40C	IPR3	IRQ 12 to 15 Priority	Read/write
0xE000E410	IPR4	IRQ 16 to 19 Priority	Read/write
0xE000E414	IPR5	IRQ 20 to 23 Priority	Read/write
0xE000E418	IPR6	IRQ 24 to 27 Priority	Read/write
0xE000E41C	IPR7	IRQ 28 to 31 Priority	Read/write
0xE000E420	IPR8	IRQ 32 to 35 Priority	Read/write
0xE000E424	IPR9	IRQ 36 to 39 Priority	Read/write
0xE000E428	IPR10	IRQ 40 to 42 Priority	Read/write

For more detail see Chapter 5, “Exceptions” and Chapter 8, “Nested Vectored Interrupt Controller” in the *ARM® Cortex™-M3 Technical Reference Manual* (DDI0337G\_cortex\_m3\_r2p0\_trm.pdf) for more information on exceptions and interrupts.

## External Interrupt Configuration

Eight external interrupts are implemented. The UHF Transceiver interrupt and these eight external interrupts can be separately configured to detect any combination of the following type of events:

1. Rising edge: The logic detects a transition from LOW to HIGH and generates a pulse. Only one pulse is sent to the Cortex-M3 per rising edge.
2. Falling edge: The logic detects a transition from HIGH to LOW and generates a pulse. Only one pulse is sent to the Cortex-M3 per falling edge.
3. Rising or falling edge: The logic detects a transition from “LOW to HIGH” or “HIGH to LOW” and generates a pulse. Only one pulse is sent to the Cortex-M3 per edge.
4. High level: The logic detects a HIGH level. The appropriate interrupt is

asserted and sent to the Cortex-M3. The interrupt line is held asserted until the external source de-asserts. The HIGH level needs to be maintained for 1 core clock cycle minimum to be detected.

5. Low level: The logic detects a LOW level. The appropriate interrupt is asserted and sent to the Cortex-M3. The interrupt line is held asserted until the external source de-asserts. The LOW level needs to be maintained for 1 core clock cycle minimum to be detected.

The external interrupt detection unit block is in the always on section and allows external interrupt to wake up the device when in HIBERNATE mode and TOTALHALT mode.

## Interrupt Memory Mapped Registers

The interrupt detection unit consists of MMRs contained in the always on block and are based at address 0x40002400.

**Table 23: Interrupt Detection Unit Memory Mapped Registers Address**  
Base Address: 0x40002400

Offset	Name	Description	Access	Default
0x0020	EI0CFG	External Interrupt configuration register 0	RW	0x0000
0x0024	EI1CFG	External Interrupt configuration register 1	RW	0x0000
0x0028	EI2CFG	UHF TRANSCEIVER interrupt configuration register	RW	0x0000
0x0030	EICLR	External interrupt clear register	RW	0x0000
0x0034	NMICLR	Non maskable interrupt clear	RW	0x0000

## External Interrupt Configuration Register 0: EI0CFG

**Table 24: EI0CFG Register Bit Description**

Address: 0x40002420

Bits	Name	Description
15	IRQ3EN	External Interrupt 3 Enable bit 1: External Interrupt 3 enabled 0: External Interrupt 3 disabled
14 to 12	IRQ3MDE	External Interrupt 3 Mode registers. 000: Rising edge 001: Falling edge 010: Rising or falling edge 011: High level 100: Low level 101: Falling edge (Same as 001) 110: Rising or falling edge (Same as 010) 111: High level (Same as 011)
11	IRQ2EN	External Interrupt 2 Enable bit 1: External Interrupt 2 enabled 0: External Interrupt 2 disabled
10 to 8	IRQ2MDE	External Interrupt 2 Mode registers. 000: Rising edge 001: Falling edge 010: Rising or falling edge 011: High level 100: Low level 101: Falling edge (Same as 001) 110: Rising or falling edge (Same as 010) 111: High level (Same as 011)
7	IRQ1EN	External Interrupt 1 Enable bit 1: External Interrupt 1 enabled 0: External Interrupt 1 disabled
6 to 4	IRQ1MDE	External Interrupt 1 Mode registers. 000: Rising edge 001: Falling edge 010: Rising or falling edge 011: High level 100: Low level 101: Falling edge (Same as 001) 110: Rising or falling edge (Same as 010) 111: High level (Same as 011)
3	IRQ0EN	External Interrupt 0 Enable bit 1: External Interrupt 0 enabled 0: External Interrupt 0 disabled

## EI0CFG register bit description continued

Bits	Name	Description
2 to 0	IRQ0MDE	External Interrupt 0 Mode registers. 000: Rising edge 001: Falling edge 010: Rising or falling edge 011: High level 100: Low level 101: Falling edge (Same as 001) 110: Rising or falling edge (Same as 010) 111: High level (Same as 011)

## External Interrupt Configuration Registers 1: EI1CFG

Table 25: EI1CFG Register Bit Description

Address: 0x40002424

Bits	Name	Description
15	IRQ7EN	External Interrupt 7 Enable bit 1: External Interrupt 7 enabled 0: External Interrupt 7 disabled
14 to 12	IRQ7MDE	External Interrupt 7 Mode registers. 000: Rising edge 001: Falling edge 010: Rising or falling edge 011: High level 100: Low level 101: Falling edge (Same as 001) 110: Rising or falling edge (Same as 010) 111: High level (Same as 011)
11	IRQ6EN	External Interrupt 6 Enable bit 1: External Interrupt 6 enabled 0: External Interrupt 6 disabled
10 to 8	IRQ6MDE	External Interrupt 6 Mode registers. 000: Rising edge 001: Falling edge 010: Rising or falling edge 011: High level 100: Low level 101: Falling edge (Same as 001) 110: Rising /falling edge (Same as 010) 111: High level (Same as 011)
7	IRQ5EN	External Interrupt 5 Enable bit 1: External Interrupt 5 enabled 0: External Interrupt 5 disabled

## EI1CFG register bit description continued

Bits	Name	Description
6 to 4	IRQ5MDE	External Interrupt 5 Mode registers. 000: Rising edge 001: Falling edge 010: Rising or falling edge 011: High level 100: Low level 101: Falling edge (Same as 001) 110: Rising /falling edge (Same as 010) 111: High level (Same as 011)
3	IRQ4EN	External Interrupt 4 Enable bit 1: External Interrupt 4 enabled 0: External Interrupt 4 disabled
2 to 0	IRQ4MDE	External Interrupt 4 Mode registers. 000: Rising edge 001: Falling edge 010: Rising or falling edge 011: High level 100: Low level 101: Falling edge (Same as 001) 110: Rising / falling edge (Same as 010) 111: High level (Same as 011)

## UHF Transceiver Interrupt Configuration Register: EI2CFG

Table 26: EI2CFG Register Bit Description

Address: 0x40002428

Bits	Name	Description
15 to 4	Reserved	Reserved
3	IRQ8EN	UHF Transceiver IRQ Enable bit 1: UHF Transceiver IRQ enabled 0: UHF Transceiver IRQ disabled
2 to 0	IRQ8MDE	UHF Transceiver IRQ Mode registers. 000: Rising edge 001: Falling edge 010: Rising or falling edge 011: High level 100: Low level 101: Falling edge (Same as 001) 110: Rising /falling edge (Same as 010) 111: High level (Same as 011)

## External interrupt Clear Register: EICLR

**Table 27: EICLR Register Bit Description**

Address: 0x40002430

Bits	Name	Description
15 to 9	Reserved	Reserved
8	IRQ8	External interrupt 8 (UHF Transceiver) clear bit. 1: clear an internal interrupt flag. Cleared automatically by hardware.
7	IRQ7	External interrupt 7 clear bit. 1: clear an internal interrupt flag. Cleared automatically by hardware.
6	IRQ6	External interrupt 6 clear bit. 1: clear an internal interrupt flag. Cleared automatically by hardware.
5	IRQ5	External interrupt 5 clear bit. 1: to clear an internal interrupt flag. Cleared automatically by hardware.
4	IRQ4	External interrupt 4 clear bit. 1: clear an internal interrupt flag. Cleared automatically by hardware.
3	IRQ3	External interrupt 3 clear bit. 1: clear an internal interrupt flag. Cleared automatically by hardware.
2	IRQ2	External interrupt 2 clear bit. 1: clear an internal interrupt flag. Cleared automatically by hardware.
1	IRQ1	External interrupt 1 clear bit. 1: clear an internal interrupt flag. Cleared automatically by hardware.
0	IRQ0	External interrupt 0 clear bit. 1: clear an internal interrupt flag. Cleared automatically by hardware.

Note: Ensure that the register write has fully completed before returning from the interrupt handler. Use the Data Synchronization Barrier (DSB) instruction if necessary.



**Non maskable interrupt Clear Register: NMICLR****Table 28: NMICLR Register Bit Description**

Address: 0x40002434

Bits	Name	Description
15 to 1	Reserved	Reserved
0	CLEAR	NMI clear bit. 1: clear an internal interrupt flag when the NMI interrupt is set. Cleared automatically by hardware.

Note: Ensure that the register write has fully completed before returning from the interrupt handler. Use the Data Synchronization Barrier (DSB) instruction if necessary.

# DMA Controller

## DMA Features

Direct memory access (DMA) is used to provide high-speed data transfer between peripherals and memory. Data can be quickly moved by DMA without any CPU actions. This keeps CPU resources free for other operations.

The DMA controller has 14 channels in total, 3 of which are Reserved. The 11 used channels are dedicated to managing DMA requests from specific peripherals. Channels are assigned as shown in Table 29.

**Table 29. DMA Channel Assignment**

Channel	Peripheral
0	SPI1 Tx
1	SPI1 Rx
2	UART Tx
3	UART Rx
4	I2C Slave Tx
5	I2C Slave Rx
6	I2C Master Tx
7	I2C Master Rx
8 to 10	Reserved
11	ADC
12	SPI0 Tx
13	SPI0 Rx

The channels are connected to dedicated hardware DMA requests; software trigger is also supported on each channel. This configuration is done by software.

Each DMA channel has a programmable priority level default or high. Within a priority level, arbitration is done using a fixed priority that is determined by the DMA channel number.

The DMA controller supports multiple DMA transfer data widths: Independent source and destination transfer size (byte, half word, and word). Source/destination addresses must be aligned on the data size.

The DMA Transfer Error interrupt is generated when a bus error condition occurs on the AHB bus during a DMA transfer.

The DMA controller supports peripheral-to-memory, memory-to-peripheral and memory-to-memory transfers and access to Flash or SRAM, as source and destination.

## DMA Operation

The DMA controller performs direct memory transfer by sharing the system bus with the Cortex™-M3 core. The DMA request may stall the CPU access to the system bus for some bus cycles, when the CPU and DMA are targeting the same destination (memory or peripheral).

For more detail see the *ARM® Cortex™-M3 Technical Reference Manual* (DDI0417A\_udma\_pl230\_r0p0\_trm.pdf) for more information on the DMA section.

### Error Management

A DMA transfer error can be generated by reading from or writing to a reserved address space. When a DMA transfer error occurs during a DMA read or a write access, the faulty channel is automatically disabled. If the DMA error interrupt is enabled in the NVIC, the error will also generate an interrupt.

### Interrupts

An interrupt can be produced when a transfer is complete for each DMA channel. Separate interrupt enable bits are available in the NVIC for each of the DMA channel.

The DMA controller fetches channel control data structures located in the SRAM memory to perform data transfers. The DMA capable peripherals, when enabled to use DMA operation request the DMA controller for transfer. At the end of the programmed number of DMA transfers for a channel, the DMA controller generates an interrupt corresponding to that channel. This interrupt indicates the completion of the DMA transfer.

### DMA Priority

The priority of a channel is determined by its number and priority level. Each channel can have two priority levels – default or high. All channels at high priority level have higher priority than all channels at default priority level. At the same priority level a channel with a lower channel number has higher priority than a channel with a higher channel number. The DMA channel priority levels can be changed by writing into the appropriate bit in the DMAPRISET register.

### Channel Control Data Structure

Every channel has two control data structures associated with it; primary data structure and an alternate data structure. For simple transfer modes, the DMA controller uses either the primary or alternate data structure. For more complex data transfer modes such as ping-pong or scatter-gather, the DMA controller uses both the primary and alternate data structures. Each control data structure (primary or alternate) occupies four 32-bit locations in the memory as shown in Table 30. The entire channel control data structure is shown in Figure 5.

**Table 30. Channel Control Data Structure**

Offset	Name	Description
0x00	SRC_END_PTR	Source End Pointer
0x04	DST_END_PTR	Destination End Pointer
0x08	CHNL_CFG	Control Data Configuration
0x0C	Reserved	Reserved

Before the controller can perform a DMA transfer, the data structure related to the DMA channel needs to be programmed at the designated location in system memory, SRAM.

- The “Source End Pointer” memory location contains the end address of the source data.
- The “Destination End Pointer” memory location contains the end address of the destination data.
- The “Control Data Configuration” memory location contains the channel configuration control data.

This determines the source and destination data size, number of transfers and the number of arbitrations etc.

	Primary Structures		Alternate Structures	
Channel 13	Reserved. Set to 0	0x0DC	Reserved. Set to 0	0x1DC
	Control	0x0D8	Control	0x1D8
	Destination End Pointer	0x0D4	Destination End Pointer	0x1D4
	Source End Pointer	0x0D0	Source End Pointer	0x1D0
Channel 12	Reserved. Set to 0	0x0CC	Reserved. Set to 0	0x1CC
	Control	0x0C8	Control	0x1C8
	Destination End Pointer	0x0C4	Destination End Pointer	0x1C4
	Source End Pointer	0x0C0	Source End Pointer	0x1C0
Channel 11	Reserved. Set to 0	0x0BC	Reserved. Set to 0	0x1BC
	Control	0x0B8	Control	0x1B8
	Destination End Pointer	0x0B4	Destination End Pointer	0x1B4
	Source End Pointer	0x0B0	Source End Pointer	0x1B0
Channel 10-8	Reserved		Reserved	
Channel 7	Reserved. Set to 0	0x07C	Reserved. Set to 0	0x17C
	Control	0x078	Control	0x178
	Destination End Pointer	0x074	Destination End Pointer	0x174
	Source End Pointer	0x070	Source End Pointer	0x170
Channel 6	Reserved. Set to 0	0x06C	Reserved. Set to 0	0x16C
	Control	0x068	Control	0x168
	Destination End Pointer	0x064	Destination End Pointer	0x164
	Source End Pointer	0x060	Source End Pointer	0x160
:	:	:	:	:
:	:	:	:	:
:	:	:	:	:
Channel 1	Reserved. Set to 0	0x01C	Reserved. Set to 0	0x11C
	Control	0x018	Control	0x118
	Destination End Pointer	0x014	Destination End Pointer	0x114
	Source End Pointer	0x010	Source End Pointer	0x110
Channel 0	Reserved. Set to 0	0x00C	Reserved. Set to 0	0x10C
	Control	0x008	Control	0x108
	Destination End Pointer	0x004	Destination End Pointer	0x104
	Source End Pointer	0x000	Source End Pointer	0x100

Figure 5: Memory Map of Primary and Alternate DMA Structures

## Control Data Configuration

For each DMA transfer, the CHNL\_CFG memory location provides the control information for the DMA Transfer to the controller.

**Table 31. Control Data Configuration**

Bits	Name	Description
31 to 30	DST_INC	<p>Destination address increment. The address increment depends on the source data width as follows:</p> <p><b>Source data width : byte</b>            00 : byte.            01 : halfword.            10 : word.            11 : no increment. Address remains set to the value that the DST_END_PTR memory location contains.</p> <p><b>Source data width : halfword</b>            00 : Reserved.            01 : halfword.            10 : word.            11 : no increment. Address remains set to the value that the DST_END_PTR memory location contains.</p> <p><b>Source data width : word</b>            00 : Reserved.            01 : Reserved.            10 : word.            11 : no increment. Address remains set to the value that the DST_END_PTR memory location contains.</p>
29 to 28	Reserved	Undefined. Write as zero.
27 to 26	SRC_INC	<p>Source address increment. The address increment depends on the source data width as follows:</p> <p><b>Source data width : byte</b>            00: byte.            01: halfword.            10: word.            11: no increment. Address remains set to the value that the SRC_END_PTR memory location contains.</p> <p><b>Source data width : halfword</b>            00: Reserved.            01: halfword.            10: word.            11: no increment. Address remains set to the value that the SRC_END_PTR memory location contains.</p> <p><b>Source data width : word</b>            00: Reserved.            01: Reserved.            10: word.            11: no increment. Address remains set to the value that the SRC_END_PTR memory location contains.</p>

## Control Data Configuration continued

Bits	Name	Description
25 to 24	SRC_SIZE	Size of the source data 00: byte 01: halfword 10: word 11: Reserved.
23 to 18	Reserved	Undefined. Write as zero.
17 to 14	R_Power	Set these bits to control how many DMA transfers can occur before the controller re-arbitrates. Needs to be set to 0000 for all DMA transfers involving peripherals. Note: the operation of the DMA will be indeterminate if a value other than "0000" is programmed in this location for DMA transfers involving peripherals.
13 to 4	N_minus_1	The number of configured transfers minus 1 for that channel. The 10-bit value indicates the number of DMA transfers(not the total number of bytes), minus one. The possible values are: 0x000: 1 DMA transfer 0x001: 2 DMA transfers 0x002: 3 DMA transfers ... 0x3FF: 1024 DMA transfers.
3	Reserved	Undefined. Write as zero.
2 to 0	Cycle_ctrl	The transfer types of the DMA cycle 000: Stop (Invalid) 001: Basic 010: Auto-request 011: Ping-Pong 100: Memory Scatter Gather Primary 101: Memory Scatter Gather Alternate 110: Peripheral Scatter Gather Primary 111: Peripheral Scatter Gather Alternate

During the DMA transfer process, before arbitration CHNL\_CFG is written back to system memory with N\_minus\_1 field changed to reflect the number of transfers yet to be done.

At the end when the whole DMA cycle is complete, the cycle\_ctrl bits are made invalid, to indicate the completion of the transfer.

## DMA Transfer Types (CHNL\_CFG [2:0])

Five types of DMA transfer are supported by DMA controller. The various types are selected by programming the appropriate values into the cycle\_ctrl bits in the CHNL\_CFG (CHNL\_CFG [2:0]) location of the control data structure.

For more detail see the *ARM® Cortex™-M3 Technical Reference Manual* (DDI0417A\_udma\_\_pl230\_r0p0\_trm.pdf) for more information on DMA transfer types.

### Invalid (CHNL\_CFG [2:0] = 000)

This means no DMA transfer is enabled for the channel. After the controller completes a DMA cycle it sets the cycle type to invalid, to prevent it from repeating the same DMA cycle.

### Basic (CHNL\_CFG [2:0] = 001)

In this mode, the controller can be configured to use either primary, or alternate data structure. The peripheral needs to present a request for every data transfer. After the channel is enabled, when the controller receives a request, it performs the following operations:

- The controller performs a transfer. If the number of transfers remaining is zero the flow continues at step 3.
- The controller arbitrates:
  - If a higher-priority channel is requesting service then the controller services that channel, or
  - If the peripheral or software signals a request to the controller then it continues at step 1.
- At the end of the transfer, the controller interrupts the processor using dma\_done interrupt for the corresponding channel.

### Auto-Request (CHNL\_CFG [2:0] = 010)

When the controller operates in this mode, it is only necessary for it to receive a single request to enable it to complete the entire DMA cycle. This allows a large data transfer to occur, without significantly increasing the latency for servicing higher priority requests, or requiring multiple requests from the processor or peripheral. This mode is very useful for a memory to memory copy application.

Auto-request is not suitable for peripheral use.

In this mode, the controller can be configured to use either primary, or alternate data structure. After the channel is enabled, when the controller receives a request, it performs the following operations.

- The controller performs,  $\min(2^{R\_power}, N)$  transfers for the channel, where R\_power is bit 17:14 of the control data configuration register and N is the number of transfers. If the number of transfers remaining is zero the flow continues at step 3.
- A request for the channel is automatically generated. The controller arbitrates. If the channel has the highest priority then the DMA cycle continues at step 1.
- At the end of the transfer, the controller generates an interrupt for the corresponding interrupt.



**Ping-pong (CHNL\_CFG [2:0] = 011)**

In ping-pong mode, the controller performs a DMA cycle using one of the data structures and then performs a DMA cycle using the other data structure. The controller continues to switch from primary to alternate to primary until it reads a data structure that is invalid, or until the host processor disables the channel.

This mode is useful for transferring data from peripheral to memory using different buffers in the memory. In a typical application, the host needs to configure both primary and alternate data structure before starting the transfer. As the transfer progresses, the host can subsequently configure primary or alternate control data structures in the interrupt service routine when corresponding transfer ends.

The DMA controller interrupts the processor after the completion of transfers associated with each control data structure. The individual transfers using either the primary or the alternate control data structures work exactly the same as a basic DMA transfer.

**Memory Scatter-Gather (CHNL\_CFG [2:0] = 100 or 101)**

In memory scatter-gather mode, the controller needs to be configured to use both primary and alternate data structure. The controller uses the primary data structure to program the control configuration for alternate data structure. The alternate data structure is used for actual data transfers a transfer similar to an Auto-Request DMA transfer. The controller arbitrates after every primary transfer. The controller only needs one request to complete the entire transfer. This mode is used when performing multiple memory to memory copy tasks. The processor can configure all of them together and is not required to intervene in between. The DMA controller interrupts the processor using the dma\_done interrupt when the entire scatter-gather transaction completes using a basic cycle.

In this mode the controller receives an initial request and then performs four DMA transfers using the primary data structure to program control structure of the alternate data structure. After this transfer completes, it starts a DMA cycle using the alternate data structure. After the cycle completes, the controller performs another four DMA transfers using the primary data structure. The controller continues to switch from primary to alternate to primary, until either:

- The processor configures the alternate data structure for a basic cycle.
- The DMA reads an invalid data structure.

Table 32 lists the fields of the CHNL\_CFG memory location for the primary data structure, which must be programmed with constant values for the memory scatter-gather mode.

**Table 32 CHNL\_CFG For Primary Data Structure in Memory Scatter Gather Mode**  
**CHNL\_CFG [2:0] = 100**

Bits	Name	Description
31 to 30	DST_INC	{10 } Configures the controller to use word increments for the address
29 to 28	Reserved	Undefined. Write as zero.
27 to 26	SRC_INC	{10 } Configures the controller to use word increments for the address
25 to 24	SRC_SIZE	{10 } Configures the controller to use word transfers
23 to 18	Reserved	Undefined. Write as zero.
17 to 14	R_power	{0010 } Indicates that the DMA controller will perform four transfers.
13 to 4	N_minus_1	Configures the controller to perform <i>N</i> DMA transfers, where <i>N</i> is a multiple of 4.
3	Reserved	Undefined. Write as zero.
2 to 0	Cycle_ctrl	{100 } Configures the controller to perform a memory scatter-gather DMA cycle

### Peripheral Scatter-Gather (CHNL\_CFG [2:0] = 110 or 111)

In peripheral scatter-gather mode, the controller needs to be configured to use both primary and alternate data structure. The controller uses the primary data structure to program the control structure of the alternate data structure. The alternate data structure is used for actual data transfers and each transfer takes place using the alternate data structure with a Basic DMA transfer. The controller does not arbitrate after every primary transfer. This mode is used when there are multiple peripheral-to-memory DMA tasks to be performed. The Cortex can configure all of them at the same time and need not intervene in between. This is very similar to memory scatter-gather mode except for arbitration and request requirements. The DMA controller interrupts the processor using the dma\_done interrupt when the entire scatter-gather transaction completes using a basic cycle.

In peripheral scatter-gather mode the controller receives an initial request from a peripheral and then performs four DMA transfers using the primary data structure to program the alternate control data structure. It then immediately starts a DMA cycle using the alternate data structure, without re-arbitrating.

After this cycle completes, the controller re-arbitrates and if it receives a request from the peripheral and has the highest priority then it performs another four DMA transfers using the primary data structure. It then immediately starts a DMA cycle using the alternate data structure without re-arbitrating. The controller continues to switch from primary to alternate to primary until either:

- The processor configures the alternate data structure for a basic cycle.
- The DMA reads an invalid data structure.

Table 33 lists the fields of the CHNL\_CFG memory location for the primary data structure, which must be programmed with constant values for the peripheral scatter-gather mode.

**Table 33 CHNL\_CFG For Primary Data Structure in Peripheral Scatter Gather Mode**  
**CHNL\_CFG [2:0] = 110**

Bits	Name	Description
31 to 30	DST_INC	{10} Configures the controller to use word increments for the address
29 to 28	Reserved	Undefined. Write as zero.
27 to 26	SRC_INC	{10} Configures the controller to use word increments for the address
25 to 24	SRC_SIZE	{10} Configures the controller to use word transfers
23 to 18	Reserved	Undefined. Write as zero.
17 to 14	R_power	{0010} Indicates that the DMA controller will perform four transfers without re arbitration.
13 to 4	N_minus_1	Configures the controller to perform <i>N</i> DMA transfers, where <i>N</i> is a multiple of four.
3	Reserved	{00} Undefined. Write as zero.
2 to 0	Cycle_ctrl	{110} Configures the controller to perform a memory scatter-gather DMA cycle

### **Address Calculation**

The DMA controller calculates the source read address based on the content of SRC\_END\_PTR, the source address increment setting in CHNL\_CFG and the current value of the N\_Minus\_1 (CHNL\_CFG {13:4}).

Similarly destination write address is calculated based on the content of DST\_END\_PTR, destination address increment setting in CHNL\_CFG and the current value of the N\_Minus\_1 (CHNL\_CFG {13:4}).

Source Read Address = SRC\_END\_PTR – (N\_minus\_1 << (SRC\_INC)) for SRC\_INC = 0,1,2

Source Read Address = SRC\_END\_PTR for SRC\_INC = 3

Destination Write Address = DST\_END\_PTR – (N\_minus\_1 << (DST\_INC)) for DST\_INC = 0,1,2

Destination Write Address = DST\_END\_PTR for DST\_INC = 3

Note: In the above equation N\_minus\_1 is the number of configured transfers minus 1 for that channel.

The ARM® PrimeCell®  $\mu$ DMA Controller Technical Reference Manual (DDI0417A\_udmac\_pl230\_r0p0\_trm.pdf) provides more information on the general operation of the DMA controller.

## DMA Memory Mapped Registers

The DMA controller interface MMRs are based at address 0x40010000.

**Table 34. DMA Controller Memory Mapped Registers Address Table**  
Base Address 0x40010000

Offset	Name	Description	Access	Default
0x0000	DMASTA	Status register	R	0X000D0000
0x0004	DMACFG	Configuration register	RW	0X00000000
0x0008	DMA PDBPTR	Primary Control Data Base Pointer register	RW	0X00000000
0x000C	DMA ADBPTR	Alternate Control Data Base Pointer register	R	0X00000100
0x0014	DMA SWREQ	Software Request register	R	0X00000000
0x0020	DMA RSKSET	Request Mask Set register	RW	0X00000000
0x0024	DMA RSKCLR	Request Mask Clear register	R	0X00000000
0x0028	DMA ENSET	Enable Set register	RW	0X00000000
0x002C	DMA ENCLR	Enable Clear register	W	0X00000000
0x0030	DMA ALTSET	Primary-Alternate Set register	RW	0X00000000
0x0034	DMA ALTCLR	Primary-Alternate Clear register	W	0X00000000
0x0038	DMA PRISET	Priority Set register	RW	0X00000000
0x003C	DMA PRICLR	Priority Clear register	W	0X00000000
0x004C	DMA ERRCLR	Bus Error Clear Register	RW	0X00000000
0x0FD0	DMA PERID4	DMA peripheral ID 4 Register	R	0X00000004
0x0FE0	DMA PERID0	DMA peripheral ID 0 Register	R	0X00000030
0x0FE4	DMA PERID1	DMA peripheral ID 1 Register	R	0X000000B2
0x0FE8	DMA PERID2	DMA peripheral ID 2 Register	R	0X0000000B
0x0FEC	DMA PERID3	DMA peripheral ID 3 Register	R	0X00000000
0x0FF0	DMA PCELLID0	DMA PrimeCell ID 0 Register	R	0X0000000D
0x0FF4	DMA PCELLID1	DMA PrimeCell ID 1 Register	R	0X000000F0
0x0FF8	DMA PCELLID2	DMA PrimeCell ID 2 Register	R	0X00000005
0x0FFC	DMA PCELLID3	DMA PrimeCell ID 3 Register	R	0X0000000B1

## DMA Status Register

**Table 35. DMASTA Register Bit Description**  
**Address:0x40001000**

Bits	Name	Description
31 to 21	Reserved	Reserved
20 to 16	CHNLSM1	Number of available DMA channels minus 1. For example if there is 14channels available, the register will read back 0XD for these bits.
15 to 8	Reserved	Reserved. Undefined.
7 to 4	STATE	Current state of DMA controller state machine. Provides insight into the operation performed by the DMA at the time this register is read.  0000: Idle 0001: Reading channel controller data 0010: Reading source data end pointer 0011: Reading destination end pointer 0100: Reading source data 0101: Writing destination data 0110: Waiting for DMA request to clear 0111: Writing channel controller data 1000: Stalled 1001: Done 1010: Peripheral scatter-gather transition 1011 to 1111: Reserved
3 to 1	Reserved	Reserved. Undefined.
0	ENABLE	Enable Status of the controller. 1: Controller is enabled 0: Controller is disabled

## DMA Configuration Register

**Table 36. DMACFG Register Bit Description**  
**Address:0x40001004**

Bits	Name	Description
31 to 1	Reserved	Reserved
0	ENABLE	Controller enable. 1: Controller is enabled 0: Controller is disabled

## DMA Primary Control Data Base Pointer Register

**Table 37: DMAPDBPTR Register Bit Description**

Address: 0x40001008

Bits	Name	Description
31 to 0	CTRLBASEPTR	<p>Pointer to the base address of the primary data structure. 5 + log (2)M LSBs are Reserved and must be written 0. M is the number of channels.</p> <p>The DMAPDBPTR register must be programmed to point to the primary channel control base pointer in the system memory. The amount of system memory that must be assigned to the DMA controller depends on the number of DMA channels used and whether the alternate channel control data structure is used. This register cannot be read when the DMA controller is in the reset state.</p>

## DMA Alternate Control Data Base Pointer Register

**Table 38: DMAADBPTR Register Bit Description**

Address: 0x4000100C

Bits	Name	Description
31 to 0	ALTCBPTR	<p>Base address of the alternate data structure.</p> <p>The DMAADBPTR read-only register returns the base address of the alternate channel control data structure. This register removes the necessity for application software to calculate the base address of the alternate data structure. This register cannot be read when the DMA controller is in the reset state.</p>

## DMA Software Request Register

Generate software request.

Set the appropriate bit to generate a software DMA request on the corresponding DMA channel. These bits are automatically cleared by the hardware after the corresponding software request completes.

**Table 39: DMASWREQ Register Bit Description**

Address: 0x40001014

Bits	Name	Description
31 to 14	Reserved	Reserved. Reads back 0.
13	SPI0RX	1: Generates a DMA request for SPI0RX 0: Does not create a DMA request for SPI0RX
12	SPI0TX	1: Generates a DMA request for SPI0TX 0: Does not create a DMA request for SPI0TX
11	ADC	1: Generates a DMA request for ADC 0: Does not create a DMA request for ADC
10 to 8	Reserved	Reserved
7	I2CMRX	1: Generates a DMA request for I2CMRX 0: Does not create a DMA request for I2CMRX
6	I2CMTX	1: Generates a DMA request for I2CMTX 0: Does not create a DMA request for I2CMTX
5	I2CSRX	1: Generates a DMA request for I2CSRX 0: Does not create a DMA request for I2CSRX
4	I2CSTX	1: Generates a DMA request for I2CSTX 0: Does not create a DMA request for I2CSTX
3	UARTRX	1: Generates a DMA request for UARTRX 0: Does not create a DMA request for UARTRX
2	UARTTX	1: Generates a DMA request for UARTTX 0: Does not create a DMA request for UARTTX
1	SPI1RX	1: Generates a DMA request for SPI1RX 0: Does not create a DMA request for SPI1RX
0	SPI1TX	1: Generates a DMA request for SPI1TX 0: Does not create a DMA request for SPI1TX

## DMA Request Mask Set Register

This register disables DMA requests from peripherals. Each bit of the register represents the corresponding channel number in the DMA controller.

Set the appropriate bit to mask the request from the corresponding DMA channel.

**Table 40: DMARMSKSET Register Bit Description**

Address: 0x40001020

Bits	Name	Description
31 to 14	Reserved	Reserved. Reads back 0.
13	SPI0RX	<p>When Read: 1: Requests are disabled for SPI0RX. 0: Requests are enabled for SPI0RX.</p> <p>When Written: 1: Disables peripheral associated with SPI0RX from generating DMA requests. 0: No effect. Use the DMARMSKCLR register to enable DMA requests.</p>
12	SPI0TX	<p>When Read: 1: Requests are disabled for SPI0TX 0: Requests are enabled for SPI0TX.</p> <p>When Written: 1: Disables peripheral associated with SPI0TX from generating DMA requests. 0: No effect. Use the DMARMSKCLR register to enable DMA requests.</p>
11	ADC	<p>When Read: 1: Requests are disabled for ADC. 0: Requests are enabled for ADC.</p> <p>When Written: 1: Disables peripheral associated with ADC from generating DMA requests. 0: No effect. Use the DMARMSKCLR register to enable DMA requests.</p>
10 to 8	Reserved	Reserved
7	I2CMRX	<p>When Read: 1: Requests are disabled for I2CMRX. 0: Requests are enabled for I2CMRX.</p> <p>When Written: 1: Disables peripheral associated with I2CMRX from generating DMA requests. 0: No effect. Use the DMARMSKCLR register to enable DMA requests.</p>
6	I2CMTX	<p>When Read: 1: Requests are disabled for I2CMTX. 0: Requests are enabled for I2CMTX.</p> <p>When Written: 1: Disables peripheral associated with I2CMTX from generating DMA requests. 0: No effect. Use the DMARMSKCLR register to enable DMA requests.</p>
5	I2CSRX	<p>When Read: 1: Requests are disabled for I2CSRX. 0: Requests are enabled for I2CSRX.</p> <p>When Written: 1: Disables peripheral associated with I2CSRX from generating DMA requests. 0: No effect. Use the DMARMSKCLR register to enable DMA requests.</p>



## DMARMSKSET Register Bit Description continued

Bits	Name	Description
4	I2CSTX	<p>When Read: 1: Requests are disabled for I2CSTX. 0: Requests are enabled for I2CSTX.</p> <p>When Written: 1: Disables peripheral associated with I2CSTX from generating DMA requests. 0: No effect. Use the DMARMSKCLR register to enable DMA requests.</p>
3	UARTRX	<p>When Read: 1: Requests are disabled for UARTRX 0: Requests are enabled for UARTRX.</p> <p>When Written: 1: Disables peripheral associated with UARTRX from generating DMA requests. 0: No effect. Use the DMARMSKCLR register to enable DMA requests.</p>
2	UARTTX	<p>When Read: 1: Requests are disabled for UARTTX. 0: Requests are enabled for UARTTX.</p> <p>When Written: 1: Disables peripheral associated with UARTTX from generating DMA requests. 0: No effect. Use the DMARMSKCLR register to enable DMA requests.</p>
1	SPI1RX	<p>When Read: 1: Requests are disabled for SPI1RX. 0: Requests are enabled for SPI1RX.</p> <p>When Written: 1: Disables peripheral associated with SPI1RX from generating DMA requests. 0: No effect. Use the DMARMSKCLR register to enable DMA requests.</p>
0	SPI1TX	<p>When Read: 1: Requests are disabled for SPI1TX. 0: Requests are enabled for SPI1TX.</p> <p>When Written: 1: Disables peripheral associated with SPI1TX from generating DMA requests. 0: No effect. Use the DMARMSKCLR register to enable DMA requests.</p>

**DMA Request Mask Clear Register**

Clear REQ\_MASK\_SET bits (CHREQMSET) in DMARMSKSET.

This register enables DMA requests from peripherals by clearing the mask set in DMARMSKSET register. Each bit of the register represents the corresponding channel number in the DMA controller.

Set the appropriate bit to clear the corresponding REQ\_MASK\_SET bits (CHREQMSET) in DMARMSKSET register.

**Table 41: DMARMSKCLR Register Bit Description**

Address: 0x40001024

Bits	Name	Description
31 to 14	Reserved	Reserved
13	SPI0RX	1: Enables peripheral associated with SPI0RX to generate DMA requests. 0: No effect. Use the DMARMSKSET register to disable DMA requests.
12	SPI0TX	1: Enables peripheral associated with SPI0TX to generate DMA requests. 0: No effect. Use the DMARMSKSET register to disable DMA requests.
11	ADC	1: Enables peripheral associated with ADC to generate DMA requests. 0: No effect. Use the DMARMSKSET register to disable DMA requests.
10 to 8	Reserved	-
7	I2CMRX	1: Enables peripheral associated with I2CMRX to generate DMA requests. 0: No effect. Use the DMARMSKSET register to disable DMA requests.
6	I2CMTX	1: Enables peripheral associated with I2CMTX to generate DMA requests. 0: No effect. Use the DMARMSKSET register to disable DMA requests.
5	I2CSRX	1: Enables peripheral associated with I2CSRX to generate DMA requests. 0: No effect. Use the DMARMSKSET register to disable DMA requests.
4	I2CSTX	1: Enables peripheral associated with I2CSTX to generate DMA requests. 0: No effect. Use the DMARMSKSET register to disable DMA requests.
3	UARTRX	1: Enables peripheral associated with UARTRX to generate DMA requests. 0: No effect. Use the DMARMSKSET register to disable DMA requests.
2	UARTTX	1: Enables peripheral associated with UARTTX to generate DMA requests. 0: No effect. Use the DMARMSKSET register to disable DMA requests.
1	SPI1RX	1: Enables peripheral associated with SPI1RX to generate DMA requests. 0: No effect. Use the DMARMSKSET register to disable DMA requests.
0	SPI1TX	1: Enables peripheral associated with SPI1TX to generate DMA requests. 0: No effect. Use the DMARMSKSET register to disable DMA requests.

## DMA Enable Set Register

Enable DMA channels.

This register allows for the enabling of DMA channels. Reading the register returns the enable status of the channels. Each bit of the register represents the corresponding channel number in the DMA controller.

Set the appropriate bit to enable the corresponding channel.

**Table 42: DMAENSET Register Bit Description**

Address: 0x40001028

Bits	Name	Description
31 to 14	Reserved	Reserved
13	SPI0RX	1: Enables SPI0RX. 0: No effect. Use the DMAENCLR register to disable the channel.
12	SPI0TX	1: Enables SPI0TX. 0: No effect. Use the DMAENCLR register to disable the channel.
11	ADC	1: Enables ADC. 0: No effect. Use the DMAENCLR register to disable the channel.
10 to 8	Reserved	-
7	I2CMRX	1: Enables I2CMRX. 0: No effect. Use the DMAENCLR register to disable the channel.
6	I2CMTX	1: Enables I2CMTX. 0: No effect. Use the DMAENCLR register to disable the channel.
5	I2CSRX	1: Enables I2CSRX. 0: No effect. Use the DMAENCLR register to disable the channel.
4	I2CSTX	1: Enables I2CSTX. 0: No effect. Use the DMAENCLR register to disable the channel.
3	UARTRX	1: Enables UARTRX. 0: No effect. Use the DMAENCLR register to disable the channel.
2	UARTTX	1: Enables UARTTX. 0: No effect. Use the DMAENCLR register to disable the channel.
1	SPI1RX	1: Enables SPI1RX. 0: No effect. Use the DMAENCLR register to disable the channel.
0	SPI1TX	1: Enables SPI1TX. 0: No effect. Use the DMAENCLR register to disable the channel.

## DMA Enable Clear Register

Disable DMA channels.

This register allows for the disabling of DMA channels. Reading the register returns the enable status of the channels. Each bit of the register represents the corresponding channel number in the DMA controller.

Note: The controller disables a channel automatically, by setting the appropriate bit, when it completes the DMA cycle.

Set the appropriate bit to disable the corresponding channel.

**Table 43: DMAENCLR Register Bit Description**

Address: 0x4000102C

Bits	Name	Description
31 to 14	Reserved	Reserved
13	SPI0RX	1: Disables SPI0RX 0: No effect. Use the DMAENSET register to enable the channel.
12	SPI0TX	1: Disables SPI0TX 0: No effect. Use the DMAENSET register to enable the channel
11	ADC	1: Disables ADC 0: No effect. Use the DMAENSET register to enable the channel
10 to 8	Reserved	-
7	I2CMRX	1: Disables I2CMRX 0: No effect. Use the DMAENSET register to enable the channel
6	I2CMTX	1: Disables I2CMTX 0: No effect. Use the DMAENSET register to enable the channel
5	I2CSRX	1: Disables I2CSRX 0: No effect. Use the DMAENSET register to enable the channel
4	I2CSTX	1: Disables I2CSTX 0: No effect. Use the DMAENSET register to enable the channel
3	UARTRX	1: Disables UARTRX 0: No effect. Use the DMAENSET register to enable the channel
2	UARTTX	1: Disables UARTTX 0: No effect. Use the DMAENSET register to enable the channel
1	SPI1RX	1: Disables SPI1RX 0: No effect. Use the DMAENSET register to enable the channel
0	SPI1TX	1: Disables SPI1TX 0: No effect. Use the DMAENSET register to enable the channel

**DMA Primary-Alternate Set Register**

Control structure status / select alt structure.

Returns the channel control data structure status, or selects the alternate data structure for the corresponding DMA channel.

Note: The DMA controller sets/clears these bits automatically as necessary for ping-pong, memory scatter-gather and peripheral scatter-gather transfers.

**Table 44: DMAALTSET Register Bit Description**

Address: 0x40001030

Bits	Name	Description
31 to 14	Reserved	Reserved
13	SPI0RX	<p>W</p> <p>When Read: 1: DMA SPI0RX is using the alternate data structure. 0: DMA SPI0RX is using the primary data structure.</p> <p>When Written: 0: No effect. Use the DMAALTCLR register to set SPI0RX to 0. 1: Selects the alternate data structure for SPI0RX</p>
12	SPI0TX	<p>When Read: 1: DMA SPI0TX is using the alternate data structure. 0: DMA SPI0TX is using the primary data structure.</p> <p>When Written: 0: No effect. Use the DMAALTCLR register to set SPI0TX to 0. 1: Selects the alternate data structure for SPI0TX</p>
11	ADC	<p>When Read: 1: DMA ADC is using the alternate data structure. 0: DMA ADC is using the primary data structure.</p> <p>When Written: 0: No effect. Use the DMAALTCLR register to set ADC to 0. 1: Selects the alternate data structure for ADC</p>
10 to 8	Reserved	-
7	I2CMRX	<p>When Read: 1: DMA I2CMRX is using the alternate data structure. 0: DMA I2CMRX is using the primary data structure.</p> <p>When Written: 0: No effect. Use the DMAALTCLR register to set I2CMRX to 0. 1: Selects the alternate data structure for I2CMRX</p>
6	I2CMTX	<p>When Read: 1: DMA I2CMTX is using the alternate data structure. 0: DMA I2CMTX is using the primary data structure.</p> <p>When Written: 0: No effect. Use the DMAALTCLR register to set I2CMTX to 0. 1: Selects the alternate data structure for I2CMTX</p>

DMAALTSET Register Bit Description continued

Bits	Name	Description
5	I2CSRX	<p>When Read: When Read:</p> <p>1: DMA I2CSRX is using the alternate data structure. 0: DMA I2CSRX is using the primary data structure.</p> <p>When Written:</p> <p>0: No effect. Use the DMAALTCLR register to set I2CSRX to 0. 1: Selects the alternate data structure for I2CSRX</p>
4	I2CSTX	<p>When Read:</p> <p>1: DMA I2CSTX is using the alternate data structure. 0: DMA I2CSTX is using the primary data structure.</p> <p>When Written:</p> <p>0: No effect. Use the DMAALTCLR register to set I2CSTX to 0. 1: Selects the alternate data structure for I2CSTX</p>
3	UARTRX	<p>When Read:</p> <p>1: DMA UARTRX is using the alternate data structure. 0: DMA UARTRX is using the primary data structure.</p> <p>When Written:</p> <p>0: No effect. Use the DMAALTCLR register to set UARTRX to 0. 1: Selects the alternate data structure for UARTRX</p>
2	UARTTX	<p>When Read:</p> <p>1: DMA UARTTX is using the alternate data structure. 0: DMA UARTTX is using the primary data structure.</p> <p>When Written:</p> <p>0: No effect. Use the DMAALTCLR register to set UARTTX to 0. 1: Selects the alternate data structure for UARTTX</p>
1	SPI1RX	<p>When Read:</p> <p>1: DMA SPI1RX is using the alternate data structure. 0: DMA SPI1RX is using the primary data structure.</p> <p>When Written:</p> <p>0: No effect. Use the DMAALTCLR register to set SPI1RX to 0. 1: Selects the alternate data structure for SPI1RX</p>
0	SPI1TX	<p>When Read:</p> <p>1: DMA SPI1TX is using the alternate data structure. 0: DMA SPI1TX is using the primary data structure.</p> <p>When Written:</p> <p>0: No effect. Use the DMAALTCLR register to set SPI1TX to 0. 1: Selects the alternate data structure for SPI1TX</p>

## DMA Primary-Alternate Clear Register

Select primary data structure.

Set the appropriate bit to select the primary data structure for the corresponding DMA channel.

Note: The DMA controller sets/clears these bits automatically as necessary for ping-pong, memory scatter-gather and peripheral scatter-gather transfers.

**Table 45: DMAALTCLR Register Bit Description**

Address: 0x40001034

Bits	Name	Description
31 to 14	Reserved	Reserved
13	SPI0RX	1: Selects the primary data structure for SPI0RX. 0: No effect. Use the DMAALTSET register to select the alternate data structure.
12	SPI0TX	1: Selects the primary data structure for SPI0TX. 0: No effect. Use the DMAALTSET register to select the alternate data structure.
11	ADC	1: Selects the primary data structure for ADC. 0: No effect. Use the DMAALTSET register to select the alternate data structure.
10 to 8	Reserved	-
7	I2CMRX	1: Selects the primary data structure for I2CMRX. 0: No effect. Use the DMAALTSET register to select the alternate data structure.
6	I2CMTX	1: Selects the primary data structure for I2CMTX. 0: No effect. Use the DMAALTSET register to select the alternate data structure.
5	I2CSRX	1: Selects the primary data structure for I2CSRX. 0: No effect. Use the DMAALTSET register to select the alternate data structure.
4	I2CSTX	1: Selects the primary data structure for I2CSTX. 0: No effect. Use the DMAALTSET register to select the alternate data structure.
3	UARTRX	1: Selects the primary data structure for UARTRX. 0: No effect. Use the DMAALTSET register to select the alternate data structure.
2	UARTTX	1: Selects the primary data structure for UARTTX. 0: No effect. Use the DMAALTSET register to select the alternate data structure.
1	SPI1RX	1: Selects the primary data structure for SPI1RX. 0: No effect. Use the DMAALTSET register to select the alternate data structure.
0	SPI1TX	1: Selects the primary data structure for SPI1TX. 0: No effect. Use the DMAALTSET register to select the alternate data structure.

## DMA Priority Set Register

Configure channel for high priority.

This register enables the user to configure a DMA channel to use the high priority level.

Reading the register returns the status of the channel priority mask. Each bit of the register represents the corresponding channel number in the DMA controller.

Returns the channel priority mask status, or sets the channel priority to high.

**Table 46: DMAPRISET Register Bit Description**

Address: 0x40001038

Bits	Name	Description
31 to 14	Reserved	Reserved
13	SPI0RX	<p>When Read:</p> <p>1: DMA SPI0RX is using a high priority level.</p> <p>0: DMA SPI0RX is using the default priority level.</p> <p>When Written:</p> <p>1: SPI0RX uses the high priority level.</p> <p>0: No effect. Use the DMAPRICLR register to set SPI0RX to the default priority level.</p>
12	SPI0TX	<p>When Read:</p> <p>1: DMA SPI0TX is using a high priority level.</p> <p>0: DMA SPI0TX is using the default priority level.</p> <p>When Written:</p> <p>1: SPI0TX uses the high priority level.</p> <p>0: No effect. Use the DMAPRICLR register to set SPI0TX to the default priority level.</p>
11	ADC	<p>When Read:</p> <p>1: DMA ADC is using a high priority level.</p> <p>0: DMA ADC is using the default priority level.</p> <p>When Written:</p> <p>1: ADC uses the high priority level.</p> <p>0: No effect. Use the DMAPRICLR register to set ADC to the default priority level.</p>
10 to 8	Reserved	-
7	I2CMRX	<p>When Read:</p> <p>1: DMA I2CMRX is using a high priority level.</p> <p>0: DMA I2CMRX is using the default priority level.</p> <p>When Written:</p> <p>1: I2CMRX uses the high priority level.</p> <p>0: No effect. Use the DMAPRICLR register to set I2CMRX to the default priority level.</p>
6	I2CMTX	<p>When Read:</p> <p>1: DMA I2CMTX is using a high priority level.</p> <p>0: DMA I2CMTX is using the default priority level.</p> <p>When Written:</p> <p>1: I2CMTX uses the high priority level.</p> <p>0: No effect. Use the DMAPRICLR register to set I2CMTX to the default priority level.</p>



## DMAPRISET Register Bit Description continued

Bits	Name	Description
5	I2CSRX	<p>When Read: 1: DMA I2CSRX is using a high priority level. 0: DMA I2CSRX is using the default priority level.</p> <p>When Written: 1: I2CSRX uses the high priority level. 0: No effect. Use the DMAPRICLR register to set I2CSRX to the default priority level.</p>
4	I2CSTX	<p>When Read: 1: DMA I2CSTX is using a high priority level. 0: DMA I2CSTX is using the default priority level.</p> <p>When Written: 1: I2CSTX uses the high priority level. 0: No effect. Use the DMAPRICLR register to set I2CSTX to the default priority level.</p>
3	UARTRX	<p>When Read: 1: DMA UARTRX is using a high priority level. 0: DMA UARTRX is using the default priority level.</p> <p>When Written: 1: UARTRX uses the high priority level. 0: No effect. Use the DMAPRICLR register to set UARTRX to the default priority level.</p>
2	UARTTX	<p>When Read: 1: DMA UARTTX is using a high priority level. 0: DMA UARTTX is using the default priority level.</p> <p>When Written: 1: UARTTX uses the high priority level. 0: No effect. Use the DMAPRICLR register to set UARTTX to the default priority level.</p>
1	SPI1RX	<p>When Read: 1: DMA SPI1RX is using a high priority level. 0: DMA SPI1RX is using the default priority level.</p> <p>When Written: 1: SPI1RX uses the high priority level. 0: No effect. Use the DMAPRICLR register to set SPI1RX to the default priority level.</p>
0	SPI1TX	<p>When Read: 1: DMA SPI1TX is using a high priority level. 0: DMA SPI1TX is using the default priority level.</p> <p>When Written: 1: SPI1TX uses the high priority level. 0: No effect. Use the DMAPRICLR register to set SPI1TX to the default priority level.</p>

## DMA Priority Clear Register

Configure channel for default priority level.

The DMAPRICLR write-only register enables the user to configure a DMA channel to use the default priority level. Each bit of the register represents the corresponding channel number in the DMA controller. Set the appropriate bit to select the default priority level for the specified DMA channel.

**Table 47: DMAPRICLR Register Bit Description**

Address: 0x4000103C

Bits	Name	Description
31 to 14	Reserved	Reserved
13	SPI0RX	1: SPI0RX uses the default priority level. 0: No effect. Use the DMAPRISET register to set SPI0RX to the high priority level.
12	SPI0TX	1: SPI0TX uses the default priority level. 0: No effect. Use the DMAPRISET register to set SPI0TX to the high priority level.
11	ADC	1: ADC uses the default priority level. 0: No effect. Use the DMAPRISET register to set ADC to the high priority level.
10 to 8	Reserved	-
7	I2CMRX	1: I2CMRX uses the default priority level. 0: No effect. Use the DMAPRISET register to set I2CMRX to the high priority level.
6	I2CMTX	1: I2CMTX uses the default priority level. 0: No effect. Use the DMAPRISET register to set I2CMTX to the high priority level.
5	I2CSRX	1: I2CSRX uses the default priority level. 0: No effect. Use the DMAPRISET register to set I2CSRX to the high priority level.
4	I2CSTX	1: I2CSTX uses the default priority level. 0: No effect. Use the DMAPRISET register to set I2CSTX to the high priority level.
3	UARTRX	1: UARTRX uses the default priority level. 0: No effect. Use the DMAPRISET register to set UARTRX to the high priority level.
2	UARTTX	1: UARTTX uses the default priority level. 0: No effect. Use the DMAPRISET register to set UARTTX to the high priority level.
1	SPI1RX	1: SPI1RX uses the default priority level. 0: No effect. Use the DMAPRISET register to set SPI1RX to the high priority level.
0	SPI1TX	1: SPI1TX uses the default priority level. 0: No effect. Use the DMAPRISET register to set SPI1TX to the high priority level.

## DMA Bus Error Clear Register

**Table 48: DMAERRCLR Register Bit Description**

Address: 0x4000104C

Bits	Name	Description
31 to 1	Reserved	Reserved
0	ERROR	<p>Bus error status.</p> <p>This register is used to read and clear the DMA bus error status. The error status is set if the controller encountered a bus error while performing a transfer. If a bus error occurs on a channel, that channel is automatically disabled by the controller. The other channels are unaffected. Write one to clear bits.</p> <p>When Read:</p> <p>1: A bus error is pending.</p> <p>0: No bus error occurred.</p> <p>When Written:</p> <p>1: Bit is cleared.</p> <p>0: No effect.</p>

## DMA Peripheral ID 4 Register

**Table 49: DMAPERID4 Register Bit Description**

Address: 0x40001FD0

Bits	Name	Description
31 to 14	Reserved	Reserved
13 to 0	PID4	DMA Peripheral ID 4. Default value 0x04

## DMA Peripheral ID 0 Register

**Table 50: DMAPERID0 Register Bit Description**

Address: 0x40001FE0

Bits	Name	Description
31 to 14	Reserved	Reserved
13 to 0	PID0	DMA Peripheral ID 0. Default value 0x30

## DMA Peripheral ID 1 Register

**Table 51: DMAPERID1 Register Bit Description**

Address: 0x40001FE4

Bits	Name	Description
31 to 14	Reserved	Reserved
13 to 0	PID1	DMA Peripheral ID 1. Default value 0xB2

## DMA Peripheral ID 2 Register

**Table 52: DMAPERID2 Register Bit Description**

Address: 0x40001FE8

Bits	Name	Description
31 to 14	Reserved	Reserved
13 to 0	PID2	DMA Peripheral ID 2 Default value 0x0B

## DMA Peripheral ID 3 Register

**Table 53: DMAPERID3 Register Bit Description**

Address: 0x40001FEC

Bits	Name	Description
31 to 14	Reserved	Reserved
13 to 0	PID3	DMA Peripheral ID 3. Default value 0x00

## DMA PrimeCell ID 0 Register

**Table 54: DMAPCELLID0 Register Bit Description**

Address: 0x40001FF0

Bits	Name	Description
31 to 14	Reserved	Reserved
13 to 0	PCCELLID0	DMA PrimeCell Identification Register 0. Default value 0x0D

## DMA PrimeCell ID 1 Register

**Table 55: DMAPCELLID1 Register Bit Description**

Address: 0x40001FF4

Bits	Name	Description
31 to 14	Reserved	Reserved
13 to 0	PCCELLID1	DMA PrimeCell Identification Register 1. Default value 0xF0

## DMA PrimeCell ID 2 Register

**Table 56: DMAPCELLID2 Register Bit Description**

Address: 0x40001FF8

Bits	Name	Description
31 to 14	Reserved	Reserved
13 to 0	PCCELLID2	DMA PrimeCell Identification Register 2. Default value 0x05

## DMA PrimeCell ID 3 Register

**Table 57: DMAPCELLID3 Register Bit Description**

Address: 0x40001FFC

Bits	Name	Description
31 to 14	Reserved	Reserved
13 to 0	PCCELLID3	DMA PrimeCell Identification Register 3. Default value 0xB1

# Flash Controller

## Flash Controller Features

The ADuCRF101 includes 128kByte of Flash/EE memory, 2kB of information space and a Flash controller.

Read & write to flash are executed by direct access.

### Commands Supported:

- Mass erase and page erase.
- Generation of signatures for single or multiple pages.
- Command abort.

Any access that the MCU core makes to the flash memory while a command is in progress will be stalled until that command completes.

### Flash Protection:

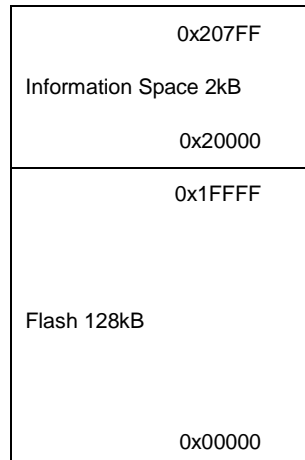
- Write protection for user space.
- Ability to lock serial wire interface for read protection.

### Flash Integrity:

- Automatic signature check of Kernel space on reset.
- User signature for application code.

## Flash Memory Organization

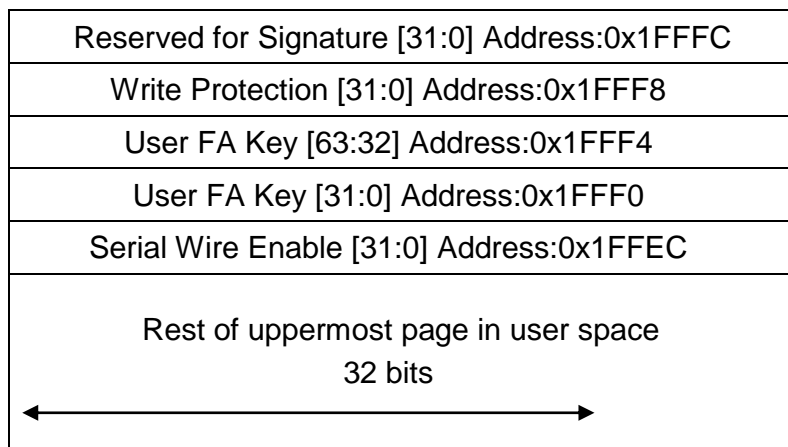
The ADuCRF101, the controller supports 128kByte of user Flash and 2kByte of information space. The information space is memory mapped above user flash space, as shown in Figure 6. The main 128kByte space is organized into 32 x flash blocks. Each flash block contains 8 flash pages – each page is 512 bytes in size. The flash is protected in block segments but it can be erased / written in page segments.



**Figure 6: Information and User Space Memory Map on ADuCRF101**

## User Space

The top 20 bytes are shown in Figure 7. If a user tries to read or write from/to a portion of memory that is not available then a bus error is returned.



**Figure 7: Uppermost Page of User Memory**

## Kernel Space

The kernel space is reserved for test data and the serial downloader and is read protected. Only 16 bytes, or 4(32 bit) locations in the kernel space are user readable. These locations are at address 0x7E8, 0x7EC, 0xFF0 and 0x7F4. Byte location 0x7E8 is used to store kernel revision in ASCII format. Location 0x7EC is not used and reads 0xFF. Location 0x7F0 and 0x7F4 are used to identify a device, as described in Table 58.

**Table 58. Silicon Identification Register Bit Description**  
Addresses 0x200007F4 and 0x200007F0

Bits	Name	Description
63 to 58	Lot ID character 6	Lot ID chars are encoded in 6 bits. 'A' to 'Z' or 'a' to 'z' -> 0 to 25 '0' to '9' -> 26 to 36 ' ' or 0xFF -> 37
57 to 52	Lot ID character 5	
51 to 46	Lot ID character 4	
45 to 40	Lot ID character 3	
39 to 34	Lot ID character 2	
33 to 28	Lot ID character 1	
27 to 22	Lot ID character 0	
21 to 16	wafer	wafer no. encoded in 6 bits -> 0 to 63
15 to 8	y	y-coordinate encoded in 8 bits -> 0 to 255 (-127 to +127)
7 to 0	x	x-coordinate encoded in 8 bits -> 0 to 255 (-127 to +127)

## Writing to Flash/EE Memory

Flash is written directly using an AHB bus write, similar to SRAM. To enable writes to the flash memory, user code must first set FEECON0 [2] to 1. Once the flash write sequence is complete as indicated by FEESTA [3], user code should then clear FEECON0 [2]. Once these bits are configured and flash protection disabled, user code may then write to the flash. When a flash erase or write operation is in progress, the Cortex-M3 core is stalled until the erase/write operation is completed.

Only 32-bit writes are supported. 16 and 8 bit writes are not supported. Burst writes to flash are supported. If the addressed location is write-protected then the controller responds with an AHB error.

If a write is followed immediately by a second write to the next location in flash and this is in the same row as the previous write then the flash write time is faster as the controller does not need to change the row address.

Note: Only an assembly instruction “store multiple” will take advantage of this flash feature.



Before performing a write to Flash/EE memory, the FEESTA register should be read and checked to ensure no other commands/writes are being executed.

## Erasing Flash/EE Memory

User code may call two flash erase commands:

- **Mass Erase.**  
Mass Erase erases the entire user flash memory. After entering the user protection key into the FEEKEY register, then write the mass erase command to FEECMD.
- **Page Erase.**  
Page Erase erases the 512byte flash page selected by FEEADR0L/FEEADR0H. After entering the User protection key into the FEEKEY register, load the FEEADR0L/FEEADR0H registers with the page address to be erased. Finally write the Page erase command to FEECMD. CMDDONE (FEESTA [2]) will indicate that the page erase command is completed.

During a Page or Mass erase sequence, the flash controller and flash block will consume extra current for the duration of the flash erase sequence. See datasheet for exact specification.

## Flash Protection

There are three types of protection implemented:

- Key-protected registers
- Read protection
- Write protection

### Flash Protection: 1. Key-protected registers

Some of the flash controller registers are key-protected to avoid accidental writes to these registers.

The user key is 0xF123F456. It is entered via the 16 bit Key register: 0xF456 first followed by 0xF123. This key must be entered to run certain user commands or write to certain locations in flash or to enable write access to the Setup register. Once entered the key remains asserted unless a command is written to the FEECMD register. When the command completes, the key will clear automatically. If this key is entered to enable write access to the Setup register or to enable writes to certain locations in flash, it needs to be cleared by user code afterwards. To clear it, write any 16 bit value to the key register.

## Flash Protection: 2. User Read Protection

User space read protection is provided by disabling serial wire access. A user can disable serial wire access by writing '0' to DBG in the flash FEECON1 register. It is also possible to disable the serial wire interface via the downloader. Serial wire access will be disabled while the kernel is running to prevent serial wire access preventing the kernel running to completion. When the kernel has completed, it will enable serial wire access by writing '1' to DBG in the flash FEECON1 register. If the serial wire enable location in the upper most page in flash is set to 0x16032010 by user code when the kernel loads it will disable serial wire access while user code is running.

## Flash Protection: 3. User Write Protection

User write protection is provided to prevent accidental writes to pages in user space and to protect blocks of user code when downloading extra code to flash. If a write or erase of a protected location is detected then the ADuCRF101 flash controller will generate an exception.

The write protection is stored at the top of user space. The top 4 bytes are reserved for a signature and the next 4 bytes below are for write protection. The write protection is uploaded by the flash controller into local registers after reset. After uploading the 32 write protection bits can be read via memory mapped register FEEPRO [31:0]. To write to the Write Protection bits a user must first write 0xF456 followed by 0xF123 to the Key register. Once the Write protection has been written it cannot be re-written without a full erase of user space. After a Mass erase the device must be reset to de-assert the uploaded copy of the write protection bits.

The following is the sequence to program the WriteProtection [31:0] bits:

- Write 0xF456 followed by 0xF123 to the key register.
- Write the required write protection directly to flash. Write '0' to enable protection. The write protection address is 0x1FFF8 for 128k bytes of flash.
- Verify that the write completed successfully by polling the status register FEESTA [3] or enabling a write complete interrupt.
- Reset the device and the WriteProtection will be uploaded from Kernel space and activated by the flash controller.

If the write protection in flash has not been programmed, that is 0xFFFFFFFF is uploaded from flash on power up, then the FEEPRO register can be written to directly from user code. This allows a user to verify the write protection before committing it to flash.

If the write protection in flash has been programmed then the MSB of the write protection must be programmed to '0' to prevent erasing of the write protection block.

For write protection memory is split into 32 blocks of 8 pages each, or 32 block of 4kBytes (1 page being 512 bytes).

If an attempt is made to write to the WriteProtection word in flash without setting the FEEKEY first then a bus error will be generated.

## Flash Controller Failure Analysis Key

It may be necessary to perform failure analysis on parts that are returned by a user but read protection is enabled. A method has been provided to allow failure analysis of protected memory: This is a user failure analysis allow key.

This key is a 64-bit key that is stored at the top of user space in flash as shown in Figure 7. It is used to gain access to user code if the serial wire interface has been locked. It is the user's responsibility to program this key to a value. The key must be given to ADI to enable access to user code.

## Flash Integrity Signature feature

The signature is used to check the integrity of the flash device. Software can call a signature check command occasionally or whenever a new block of code is about to be executed. The signature is a 24 bit CRC with the polynomial  $x^{24} + x^{23} + x^6 + x^5 + x + 1$ .

The Sign command can be used to generate a signature and check the signature of a block of code, where a block can be a single page or multiple pages. A 24 bit LFSR is used to generate the signature. The hardware assumes that the signature for a block is stored in the upper 4 bytes of the most significant page of a block; therefore these 4 bytes will not be included when generating the signature.

The following procedure should be followed to generate a signature:

- Write the start address of the block to FEEADR0L/H register.
- Write the end address of the block to the FEEADR1L/H register.
- Write the Sign command to the command register. (FEECMD = 0x2)

When the command has completed the signature is available in the Sign register. The signature is compared with the data stored in the upper 4 bytes of the uppermost page of the block. If the data does not match the signature, a fail status is returned in the status register (FEESTA [5:4] = 10).

While the signature is being computed all other accesses to flash are stalled. For a 128k byte block that is 32k reads.

Note that FEEADR0L/H and FEEADR1L/H addresses are byte addresses but only pages need to be identified – that is the lower 9 bits are ignored by the hardware.

Note that the user must run the CRC polynomial in user code first to generate the CRC value and must write this to the upper 4 bytes of the upper most page of a block. Once this operation is complete, any call of the Signature feature will compare this 4 byte value to the result of the signature check function.

## Integrity of the Kernel

The hardware automatically checks the integrity of the kernel after reset. In the event of a failure, FEESTA [6] is set and user code will not run. This bit can only be read via a serial wire read, if the serial wire interface is enabled.

## Abort Using Interrupts

Commands (erase, sign or mass verify) and writes can be aborted on receipt of an interrupt as listed in Table 21. Aborts are also possible by writing an abort command to the FEECMD register. However if flash is being programmed and the routine controlling the programming is in flash then it is not possible to use the Abort command to abort the cycle as instructions cannot be read. Therefore the ability to abort a cycle on the assertion of any system interrupt is provided. The FEEAENx register is used to enable aborts on receipt of an interrupt.

When a command or write is aborted via a system interrupt FEESTA [5:4] will indicate 'aborted' (FEESTA [5:4] = 11).

It is not possible to abort a flash write or erase cycle without waiting for the high voltage in the flash core to discharge first. That is a write cycle must finish with by waiting time 6.6µs for the high voltage to discharge. An erase cycle must finish with a waiting time of 6.6µs. A mass erase cycle must finish with a waiting time of 121µs.

Depending on the state a write cycle is in, when abort asserts the Write cycle may or may not complete. If the write or erase cycle did not complete successfully then a fail status of aborted can be read in the status register.

If an immediate response to an interrupt is required during an erase or program cycle then the interrupt service routine and the 'interrupt vector table' must be moved to SRAM for the duration of the cycle.

If the DMA engine is setup to write a block of data to Flash then an interrupt can be setup to abort the current write but the DMA engine will start the next write immediately. The interrupt causing the abort will stay asserted so there will be a number of aborted write cycles in this case before the processor gets access to flash.

When an abort is triggered by an interrupt all commands will be repeatedly aborted until the appropriate FEEAENx bit is cleared or the interrupt source is cleared.

## Flash Controller Performance and Command Duration

Direct single Write access (32-bit location): 46µs.

Mass erase: 21ms.

Page erase: 21ms.

Direct write of a page (128 32-bit locations): 3.04ms.

MassVerify for all of user space: 2.05ms (128k bytes and 16MHz HCLK).

Sign for all of user space: 2.05ms (128k bytes and 16MHz HCLK).

Note: These flash timings are not dependant on the clock dividers as shown in Figure 3.

# Flash Controller Memory Mapped Registers

**Table 59. Flash Controller Memory Mapped Registers Address Table**  
Base Address 0x40002800

Offset	Name	Description	Access	Default
0x0000	FEESTA	Status register	R	0x0000
0x0004	FEECON0	Command Control register	RW	0x0000
0x0008	FEECMD	Command register	RW	0x0000
0x0010	FEEADR0L	Lower page address register	RW	0x0000
0x0014	FEEADR0H	Upper page address register	RW	0x0000
0x0018	FEEADR1L	Lower page address register	RW	0x0000
0x001C	FEEADR1H	Upper page address register	RW	0x0000
0x0020	FEEKEY	The key register	W	0x0000
0x0028	FEEPROL	Lower 16 bits of the write protection register	RW	0xFFFF
0x002C	FEEPROH	Upper 16 bits of the write protection register	RW	0xFFFF
0x0030	FEESIGL	Lower 16 bits of the signature	R	Modified by the Kernel
0x0034	FEESIGH	Upper 16 bits of the signature	R	Modified by the Kernel
0x0038	FEECON1	User Setup register	RW	Modified by the Kernel
0x0048	FEEADRAL	Lower 16 bits of the write abort address register	R	0x0800
0x004C	FEEADRAH	Upper 16 bits of the write abort address register	R	0x0002
0x0078	FEEAEN0	Interrupt abort enable register. Interrupts 15 to 0	RW	0x0000
0x007C	FEEAEN1	Interrupt abort enable register. Interrupts 31 to 16	RW	0x0000
0x0080	FEEAEN2	Interrupt abort enable register. Interrupts 33 to 32	RW	0x0000

## Flash Memory Status Register

Table 60: FEESTA Register Bit Description

Address: 0x40002800

Bits	Name	Description															
15 to 7	Reserved	Return 0 when read.															
6	SIGNERR	Kernel space signature check on reset error. After reset the flash controller automatically checks the Information space signature. If the signature check fails this bit will asserted. User can check if this bit is set via serial wire only. User code does not execute if this bit is set.															
5 to 4	CMDRES	<div>These two bits indicate the status of a command on completion or the status of a write via the AHB bus. If multiple commands are executed or there are multiple writes via the AHB bus without a read of the status register then the first error encountered is stored.</div> <table><tr><th>Bits</th><th></th><th>Description</th></tr><tr><td>00</td><td></td><td>Indicates a successful completion of a command or a write</td></tr><tr><td>01</td><td></td><td>Indicates an attempted erase of a protected location</td></tr><tr><td>10</td><td></td><td>Indicates a Read verify error. After an erase the controller reads the corresponding word(s) to verify that the transaction completed successfully. If data read is not all 'F's this is the resulting status. If the Sign command is executed and the resulting signature does not match the data in the upper 4 bytes of the upper page in a block then this is the resulting status.</td></tr><tr><td>11</td><td></td><td>Indicates that a command or a write was aborted by an abort command or a system interrupt has caused an abort.</td></tr></table> <div>These bits clear to '00' when read.</div>	Bits		Description	00		Indicates a successful completion of a command or a write	01		Indicates an attempted erase of a protected location	10		Indicates a Read verify error. After an erase the controller reads the corresponding word(s) to verify that the transaction completed successfully. If data read is not all 'F's this is the resulting status. If the Sign command is executed and the resulting signature does not match the data in the upper 4 bytes of the upper page in a block then this is the resulting status.	11		Indicates that a command or a write was aborted by an abort command or a system interrupt has caused an abort.
Bits		Description															
00		Indicates a successful completion of a command or a write															
01		Indicates an attempted erase of a protected location															
10		Indicates a Read verify error. After an erase the controller reads the corresponding word(s) to verify that the transaction completed successfully. If data read is not all 'F's this is the resulting status. If the Sign command is executed and the resulting signature does not match the data in the upper 4 bytes of the upper page in a block then this is the resulting status.															
11		Indicates that a command or a write was aborted by an abort command or a system interrupt has caused an abort.															
3	WRDONE	Write Complete Set to 1 when a write via the AHB bus completes. Cleared to 0 when read. If there are multiple writes (or a burst write), this status bit will assert after the first long word written and will stay asserted until read. If there is a burst write to flash then this bit will assert after every long word written (assuming the bit is cleared by a read after every long word written).															
2	CMDDONE	Command complete Set to 1 when a command completes. Cleared to 0 when read. If there are multiple commands, this status bit will assert after the first command completes and will stay asserted until read.															
1	WRBUSY	Write busy Set to 1 when the flash block is executing a write via the AHB bus.															
0	CMDBUSY	Command busy Set to 1 when the flash block is executing any command entered via the command register															

## Flash Memory Control Register

**Table 61: FEECON0 Register Bit Description**  
Address: 0x40002804

Bits	Name	Description
15 to 3	Reserved	Return 0 when read
2	WREN	Write enable 1: Enable writing to the flash. 0: Disable writing to the flash. A flash write when this bit is set to 1 will result in an AHB bus error and the write will not take place.
1	IENERR	Error interrupt enable. 1: An interrupt will be generated when a command or flash write completes with an error status
0	IENCMD	Command complete interrupt enable. 1: An interrupt will be generated when a command or flash write completes

## Flash Memory Command Register

**Table 62: FEECMD Register Bit Description**  
Address: 0x40002808

Bits	Name	Description
15 to 4	Reserved	Return 0 when read
3 to 0	CMD	Commands

The following commands are supported by the flash block.

For repeated page erase commands the key must be entered before each command. If a command is entered without entering the key first then no action will be taken, CMDDONE will not assert.

Any core access to the flash is stalled until the command completes.

Table 63: Flash Controller Commands (FEECMD[3:0])

FEECMD[3:0]	Name	Description
0000	IDLE	No command executed
0001	ERASEPAGE	<p>Write the address of the page to be erased to FEEADR0L/H, then write this code to the FEECMD register and the flash will erase the page. When the erase has completed the flash will read every location in the page to verify all words in the page are erased. If there is a read verify error this will be indicated in FEESTA.</p> <p>To erase multiple pages wait until a previous page erase has completed – check the status then issue a command to start the next page erase.</p> <p>Before entering this command 0xF456 followed by 0xF123 must be written to the Key register.</p>
0010	SIGN	<p>Use this command to generate a signature for a block of data. The signature is generated on a page by page basis. To generate a signature the address of the first page of the block is entered in FEEADR0L/H, the address of the last page is written to FEEADR1L/H, then write this code to the FEECMD register. When the command has complete the signature is available for reading in FEESIGL/H.</p> <p>The last 4 bytes of the last page in a block is Reserved for storing the signature.</p> <p>Before entering this command 0xF456 followed 0xF123 must be written to the Key register.</p>
0011	MASSERASE	<p>Erase all of user space. To enable this operation 0xF456 followed by 0xF123 must first be written to FEEKEY (this is to prevent accidental erases).</p> <p>When the mass erase has completed the controller will read every location to verify that all locations are 0xFFFFFFFF. If there is a read verify error this will be indicated in FEESTA.</p>
0100	ABORT	<p>If this command is issued then any command currently in progress will be stopped. The status will indicate command completed with an error status in FEESTA[5:4]. Note that this is the only command that can be issued while another command is already in progress. This command can also be used to stop a write that may be in progress.</p> <p>If a write is aborted the address of the location been written can be read via the FEEADR0L/H register. While the flash controller is writing one longword another longword write may be in the pipeline from the Cortex or DMA engine (depending on how the software implements writes). Therefore both writes may need to be aborted.</p> <p>If a write or erase is aborted then the flash timing will be violated and it is not possible to determine if the write or erase completed successfully.</p> <p>To enable this operation 0xF456 followed by 0xF123 must first be written to FEEKEY (this is to prevent accidental aborts).</p>



### Flash Controller Page Address 0 Registers

For an erase command it indicates the page address to be erased

For a sign command it indicates the start page for calculating the signature

**Table 64: FEEADR0L Register Bit Description**

Address: 0x40002810

Bits	Name	Description
15 to 9	VALUE	Bits 15 to 9 of the address of a page in flash. Used by the Erase and Sign commands
8 to 0	Reserved	9 Reserved bits for byte addresses The lower 9 bits of a byte address are ignored here as erase and sign commands use page address. Returns 0x0 if read.

**Table 65: FEEADR0H Register Bit Description**

Address: 0x40002814

Bits	Name	Description
15 to 1	Reserved	Set to 0
0	VALUE	Bit 16 of the page address.

### Flash Controller Page Address 1 Registers

For a sign command it indicates the end page for calculating the signature

**Table 66: FEEADR1L Register Bit Description**

Address: 0x40002818

Bits	Name	Description
15 to 9	VALUE	Bits 15 to 9 of the address of a page in flash. Used to identify the last page used by the Sign command.
8 to 0	Reserved	9 Reserved bits for byte addresses The lower 9 bits of a byte address are ignored here as the Sign command uses page address. Returns 0x0 if read.

**Table 67: FEEADR1H Register Bit Description**

Address: 0x4000281C

Bits	Name	Description
15 to 1	Reserved	Set to 0
0	VALUE	Bit 16 of the page address.

## Flash Controller Key Register

Table 68: FEEKEY Register Bit Description

Address: 0x40002820

Bits	Name	Description
15 to 0	VALUE	Enter 0xF456 followed by 0xF123. Returns 0x0 if read.

## Flash Controller Write Protection Registers

Table 69: FEEPROL Register Bit Description

Address: 0x40002828

Bits	Name	Description
15 to 0	VALUE	Lower 16 bits of the Write protection. 1: Leave a flash block unprotected 0: Protect a section of flash. This register is read only if the write protection in flash has been programmed.

Table 70: FEEPROH Register Bit Description

Address: 0x4000282C

Bits	Name	Description
15 to 0	VALUE	Upper 16 bits of the Write protection. 1: Leave a flash block unprotected 0: Protect a section of flash. This register is read only if the write protection in flash has been programmed.

## Flash Controller Signature Registers

Table 71: FEESIGL Register Bit Description

Address: 0x40002830

Bits	Name	Description
15 to 0	VALUE	Lower 16 bits of the signature. Signature[15:0].

Table 72: FEESIGH Register Bit Description

Address: 0x40002834

Bits	Name	Description
15 to 8	Reserved	
7 to 0	VALUE	Upper 8 bits of the signature. Signature[23:16].

## Flash Controller Setup Register

FEECON1 register is key-protected. The key must be entered in FEEKEY. After writing to FEECON1, a 16-bit value must be written again to FEEKEY, to re-assert the key protection.

**Table 73: FEECON1 Register Bit Description**

Address: 0x40002838

Bits	Name	Description
15 to 1	Reserved	Returns 0 when read
0	DBG	<p>Serial wire debug enable:            1: Enable access via the serial wire debug interface            0: Disable access via the serial wire debug interface</p> <p>The kernel set this bit to '1' when it has finished executing, thus enabling debug access to a user.</p> <p>If the Serial Wire Enable location in the last page of flash is set to 0x16032010 by user code when the kernel loads it will disable serial wire access while user code is running therefore the kernel sets this bit to '0' when it has finished executing.</p> <p>If the Serial Wire Enable location is not set to 0x16032010 by user code when the kernel loads it will enable serial wire access while user code is running therefore the kernel sets this bit to '1' when it has finished executing.</p>

## Flash Controller Write Abort Address Registers

**Table 74: FEEADRAL Register Bit Description**

Address: 0x40002848

Bits	Name	Description
15 to 0	VALUE	<p>Lower 16 bits of the FEEADRA register.</p> <p>If a write is aborted then this will contain the address of the location been written when the write was aborted.</p>

**Table 75: FEEADRAH Register Bit Description**

Address: 0x4000284C

Bits	Name	Description
15 to 0	VALUE	Upper 16 bits of the FEEADRA register

## Flash Controller System IRQ Abort Enable Registers

Table 77: FEEAENX Register Bit Description

FEEAEN0 Address: 0x40002878, FEEAEN1 Address: 0x4000287C, FEEAEN2 Address: 0x40002880,

Bits	Name	Description
15 to 0	VALUE	<p>To allow a system interrupt to abort a write or a command (erase, sign or mass verify) then write a '1' to the appropriate bit in this register.</p> <p>The appropriate bit is determined by the interrupt required to abort the flash command. for example, if external IRQ2 is required to abort a flash command, then set FEEAEN0 = 0x8.</p> <p>FEEAEN0[15:0] enable interrupts 15 to 0 to abort the flash operation  FEEAEN1[31:16] enable interrupts 31 to 16 to abort the flash operation  FEEAEN2[42:32] enable interrupts 42 to 32 to abort the flash operation  Must correspond with the number of interrupts</p>

# Reset

## Reset Operation

Reset is available on the Top and bottom die. The top die reset (ADF7023) is discussed in the UHF transceiver section. This chapter describes the reset for the bottom die.

There are four kinds of resets on the bottom die:

1. External Reset
2. Power-on Reset
3. Watchdog timeout
4. Software System Reset.

The software system reset is provided as part of the Cortex core. To generate a software system reset, the Application interrupt/Reset Control Register must be written to 0x05FA0004. This register is part of the NVIC register and is located at address 0xE000ED0C. See *ARM® Cortex™-M3 Technical Reference Manual* for more details on the software reset.

The RSTSTA register stores the reset cause until it is cleared by writing the RSTCLR register. These registers can be used during a reset exception service routine to identify the source of the reset.

**Table 78: Device Reset Implications**

Reset	Impact					
	Reset external pins to default state	Execute kernel	Reset all MMRs except RSTSTA	Reset all peripherals	Valid SRAM	RSTSTA after reset event
SW	YES <sup>1</sup>	YES	YES	YES	YES/NO <sup>2</sup>	RSTSTA[3] = 1
WD	YES <sup>1</sup>	YES	YES	YES	YES/NO <sup>2</sup>	RSTSTA[2] = 1
External reset pin	YES <sup>1</sup>	YES	YES	YES	YES/NO <sup>2</sup>	RSTSTA[1] = 1
POR	YES <sup>1</sup>	YES	YES	YES	NO	RSTSTA[0] = 1

<sup>1</sup> GPIO x will return to its default state i.e. POR output. It will only be low in the case of a POR event; in all other cases it will remain high.

<sup>2</sup> RAM is not valid in the case of a reset following a UART download.

## Reset Memory Mapped Registers

**Table 79: Reset Memory Mapped Register Address Table**

Base address: 0x40002400

Offset	Name	Description	Access	Default
0x0040	RSTSTA	Status register, read only.	R	0x01, depends on the type of reset
0x0040	RSTCLR	Clear Status register, write only.	W	N/A

### Reset Status/ Status Clear Registers

**Table 80: RSTSTA/CLR Register Bit Description**

Base address: 0x40002440

Bits	Name	Description
7 to 4	Reserved	Reserved
3	SWRST	Software reset. 1: Set automatically to 1 when the Cortex system reset is generated. 0: Cleared by setting the corresponding bit in RSTCLR.
2	WDRST	Watchdog timeout. 1: Set automatically to 1 when a watchdog timeout occurs. 0: Cleared by setting the corresponding bit in RSTCLR.
1	EXTRST	External reset. 1: Set automatically to 1 when an external reset occurs. 0: Cleared by setting the corresponding bit in RSTCLR.
0	POR	Power-on reset. 1: Set automatically when a power-on reset occurs. 0: Cleared by setting the corresponding bit in RSTCLR.

# UHF Transceiver

## UHF Transceiver Features

The ADuCRF101 is a two-die solution, integrating the ADF7023, stand alone UHF transceiver, as the top die.

ADF7023 short-range RF transceiver is a low-power low-IF transceiver designed for operation in the 862 MHz to 928 MHz and 431 MHz to 464 MHz frequency bands, which cover the worldwide license-free ISM bands at 433MHz, 868MHz and 915 MHz

The ADF7023 includes enhanced baseband features and offers a low transmit-and-receive current, as well as data rates in 2FSK/GFSK up to 300 kbps. It offers industry-leading blocking resistance and receive sensitivity with very low receive and transmit currents, allowing the radio to operate in environments where interfering signals are greater than four times stronger and more than double the operating range. It operates with a power supply range of 1.8 V to 3.6 V and has lower power consumption in both transmit and receive modes, enabling longer battery life.

Other features integrated on-chip include an extremely low-power, 8-bit RISC communications processor; patent-pending, fully-integrated image rejection scheme; VCO (voltage controlled oscillator); Fractional-N PLL (phase locked loop); 8-bit ADC (analog-to-digital converter); digital RSSI (received signal strength indication); temperature sensors; AFC (automatic frequency control) loop; and battery voltage monitor.

For more information on the ADF7023, please visit the ADI website.

In the context of this chapter the host processor is the Cortex-M3 processor.

## UHF Transceiver Simplified Block Diagram

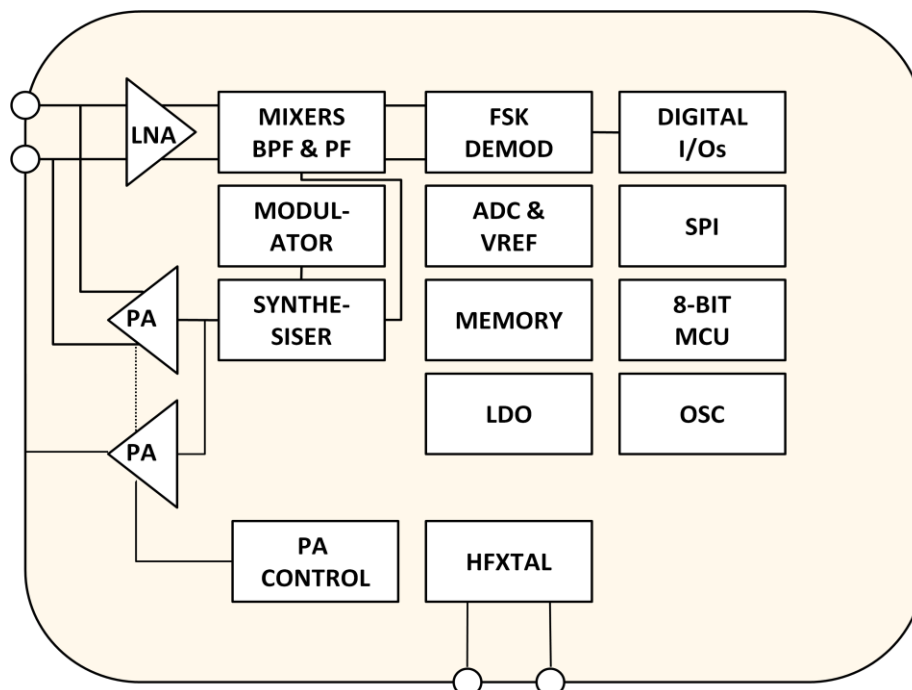


Figure 8: UHF Transceiver Simplified Block Diagram

# Radio Control

## Operational States

The UHF transceiver has five radio states:

1. PHY\_SLEEP
2. PHY\_OFF
3. PHY\_ON
4. PHY\_RX
5. PHY\_TX

The host processor can transition the UHF transceiver between states by issuing single byte commands over the SPI interface. The various commands and states are illustrated in **Figure 9**. The communications processor handles the sequencing of various radio circuits and critical timing thereby simplifying radio operation and easing the burden on the host processor.

### PHY\_SLEEP

In this state the device is in a low power sleep mode. To enter the state the command CMD\_PHY\_SLEEP should be issued, either from the PHY\_OFF or PHY\_ON states. To wake the radio from the state the CS pin can be set low, or the wake up controller (32.768 kHz RC or 32.768 kHz crystal) can be used to wake the radio from this state. The wake up timer should be setup before entering the PHY\_SLEEP state. If retention of BBRAM contents is not required then Deep Sleep Mode 2 can be used to further reduce the PHY\_SLEEP current consumption. Deep Sleep Mode 2 is entered by issuing the CMD\_HW\_RESET command. The options for PHY\_SLEEP state are detailed in Table 81.

### PHY\_OFF

In this state the 26 MHz crystal, the digital regulator and the synthesizer regulator are powered up. All memories are fully accessible. The BBRAM registers must be valid before exiting this state.

### PHY\_ON

In this state, along with the crystal, the digital regulator and the synthesizer regulator, the VCO and RF regulators are powered up. A baseband filter calibration is performed when this state is entered from the PHY\_OFF state if the BB\_CAL bit in the mode\_control register (Address 0x11A) is set. The device is ready to operate and the PHY\_TX and PHY\_RX states can be entered.

### PHY\_TX

In this state the synthesizer is enabled and calibrated. The power amplifier is enabled and the device transmits at the channel frequency defined by the CHANNEL\_FREQ [23:0] setting (Address 0x109 to Address 0x10B). The state is entered by issuing the



CMD\_PHY\_TX command. The device automatically transmits the transmit packet stored in the packet RAM. After transmission of the packet, the PA is disabled and the device automatically returns to the PHY\_ON state and can, optionally, generate an interrupt.

In SPORT mode the device transmits the data present on the GP2 pin. The host processor must issue the CMD\_PHY\_ON command to exit PHY\_TX when in SPORT mode.

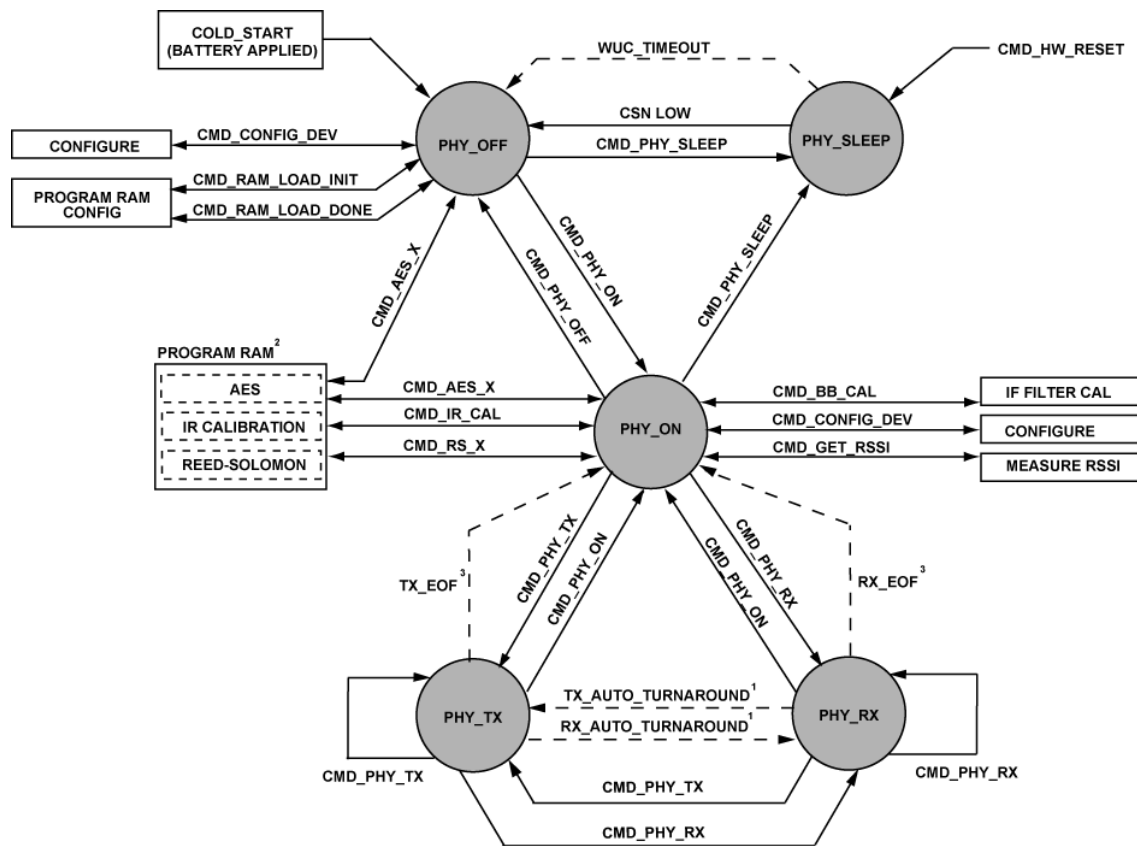
## PHY\_RX

In this state, the synthesizer is enabled and calibrated. The ADC, RSSI, IF filter, mixer, and LNA are enabled. The radio is in receive mode on the channel frequency defined by the CHANNEL\_FREQ [23:0] setting (Address 0x109 to Address 0x10B). After reception of a valid packet, the device returns to the PHY\_ON state and can, optionally, generate an interrupt. In sport mode, the device remains in the PHY\_RX state until the CMD\_PHY\_ON command is issued.

**Table 81: Sleep Modes Current Consumption**

State	Current (typical)	Conditions
PHY_SLEEP (deep sleep mode 2)	0.18uA	Wake up timer off, BBRAM contents not retained. Entered by issuing CMD_HW_RESET
PHY_SLEEP (deep sleep mode 1)	0.33uA	Wake up timer off, BBRAM contents retained
PHY_OFF	1.0mA	
PHY_ON	1.0mA	
PHY_TX	24.1mA	10dBm, single-ended PA, 868MHz
PHY_RX	12.8mA	

The state diagram of the communications processor is detailed in Figure 9.



1. Transmit and receive automatic turnaround must be enabled by bits rx\_auto\_turnaround and tx\_auto\_turnaround (0x11A: mode\_control)
2. AES encryption/decryption and image rejection calibration (IR cal) are only available if the necessary firmware module has been downloaded to the program RAM
3. The end of frame (EOF) automatic transitions are disabled in SPORT mode

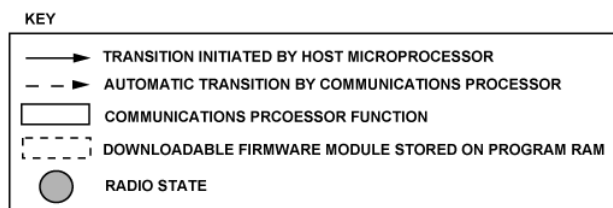


Figure 9: Radio Controller State Diagram for Normal Operation

## Initialization

### Initialization after Application of Power

When power is applied to the UHF transceiver ( Through the VDDBAT1/ VDDBAT2 pins), it registers a power on reset event (POR) and transitions to the PHY\_OFF state. The BBRAM memory is unknown, the packet RAM memory is cleared to 0x00 and the MCR memory is reset to its default values. The following procedure should be used by the host processor to complete the initialization sequence:

1. Bring the  $\overline{CS}$  line of the SPI low and wait until the MISO output goes high.
2. Issue the CMD\_SYNC command.
3. Wait for the CMD\_READY bit in the status word to go high.
4. Configure the part by writing to all 64 of the BBRAM registers.
5. Issue the CMD\_CONFIG\_DEV command so that the radio and packet handler settings are updated using the BBRAM values.
6. The UHF transceiver is now configured and ready to transition to PHY\_ON.

### Initialization after Issuing the CMD\_HW\_RESET Command

The CMD\_HW\_RESET command performs a full power down of all hardware and the device enters the PHY\_SLEEP state. To complete the hardware reset the host processor should complete the following procedure:

1. Wait for 1ms
2. Bring the  $\overline{CS}$  line of the SPI low and wait until the MISO output goes high. The UHF transceiver registers a POR and enters the PHY\_OFF state.
3. Issue the CMD\_SYNC command.
4. Wait for the CMD\_READY bit in the status word to go high.
5. Configure the part by writing to all 64 of the BBRAM registers.
6. Issue the CMD\_CONFIG\_DEV command so that the radio are updated using the BBRAM values.
7. The UHF transceiver is now configured and ready to transition to PHY\_ON.

### Initialization on Transitioning from PHY\_SLEEP (After $\overline{CS}$ Is Brought Low)

The host processor can bring  $\overline{CS}$  low at any time in order to wake the UHF transceiver from the PHY\_SLEEP state. This event is not registered as a POR event as the BBRAM contents are valid. The following is the procedure that the host processor is required to follow:

1. Bring the  $\overline{CS}$  line of the SPI low and wait until the MISO output goes high. The UHF transceiver enters PHY\_OFF.
2. Issue the CMD\_SYNC command.
3. Wait for the CMD\_READY bit in the status word to go high.
4. Issue the CMD\_CONFIG\_DEV command so that the radio are updated using the BBRAM values.
5. The UHF transceiver is now configured and ready to transition to PHY\_ON.

## Initialization after WUC Timeout

The UHF transceiver can autonomously wake from the PHY\_SLEEP state using the wake-up controller. If the UHF transceiver wakes after a WUC timeout in smart wake mode (SWM) it follows the SWM routine based on the smart wake mode configuration in BBRAM. If the UHF transceiver wakes after a WUC timeout, with SWM disabled and the firmware timer disabled, it wakes in PHY\_OFF state.

The following is the procedure that the host processor is required to follow:

1. Issue the CMD\_SYNC command.
2. Wait for the CMD\_READY bit in the status word to go high.
3. Issue the CMD\_CONFIG\_DEV command so that the radio are updated using the BBRAM values.
4. The UHF transceiver is now configured and ready to transition to PHY\_ON.

## Commands

The commands that are supported by the radio controller are detailed in Table 82. They initiate transitions between radio states or perform tasks as indicated in Figure 9.

**Table 82: Radio Controller Commands**

Commands	Code	Description
CMD_SYNC	0xA2	Synchronizes the communications processor to the host processor after reset.
CMD_PHY_OFF	0xB0	Performs a transition of the device into the PHY_OFF state.
CMD_PHY_ON	0xB1	Performs a transition of the device into the PHY_ON state.
CMD_PHY_RX	0xB2	Performs a transition of the device into the PHY_RX state.
CMD_PHY_TX	0xB5	Performs a transition of the device into the PHY_TX state.
CMD_PHY_SLEEP	0xBA	Performs a transition of the device into the PHY_SLEEP state.
CMD_CONFIG_DEV	0xBB	Configures the radio parameters based on the BBRAM values.
CMD_GET_RSSI	0xBC	Performs an RSSI measurement.
CMD_BB_CAL	0xBE	Performs a calibration of the IF filter.
CMD_HW_RESET	0xC8	Performs a full hardware reset. The device enters the PHY_SLEEP state.
CMD_RAM_LOAD_INIT	0xBF	Prepares the program RAM for a firmware module download.
CMD_RAM_LOAD_DONE	0xC7	Performs a reset of the communications processor after download of a firmware module to program RAM.
CMD_IR_CAL <sup>1</sup>	0xBD	Initiates an image rejection calibration routine.
CMD_AES_ENCRYPT <sup>2</sup>	0xD0	Performs an AES encryption on the transmit payload data stored in packet RAM.
CMD_AES_DECRYPT <sup>2</sup>	0xD2	Performs an AES decryption on the received payload data stored in packet RAM.
CMD_AES_DECRYPT_INIT <sup>2</sup>	0xD1	Initializes the internal variables required for AES decryption.
CMD_RS_ENCODE_INIT <sup>3</sup>	0XD1	Initializes the internal variables required for the Reed Solomon encoding.
CMD_RS_ENCODE <sup>3</sup>	0XD0	Calculates and appends the Reed Solomon check bytes to the transmit payload data stored in packet RAM.
CMD_RS_DECODE <sup>3</sup>	0xD2	Performs a Reed Solomon error correction on the received payload data stored in packet RAM.

<sup>1</sup> The image rejection calibration firmware module must be loaded to program RAM for this command to be functional.

<sup>2</sup> The AES firmware module must be loaded to program RAM for this command to be functional.

<sup>3</sup> The Reed Solomon Coding firmware module must be loaded to program RAM for this command to be functional.

**CMD\_PHY\_OFF (0xB0)**

This command transitions the UHF transceiver to the PHY\_OFF state. It can be issued in the PHY\_ON state. It powers down the RF and VCO regulators.

**CMD\_PHY\_ON (0xB1)**

This command transitions the UHF transceiver to the PHY\_ON state. If the command is issued in the PHY\_OFF state it powers up the RF and VCO regulators and performs an IF filter calibration if the BB\_CAL bit is set in the MODE\_CONTROL REGISTER (Address 0x11A).

If the command is issued from the PHY\_TX state, the communications processor performs the following procedure:

1. Ramps down the PA
2. Sets the external PA signal low (if enabled)
3. Turns off the digital transmit clocks
4. Powers down the synthesizer
5. Sets FW\_STATE = PHY\_ON

If the command is issued from PHY\_RX state, the communications processor performs the following procedure

1. Copies the measured RSSI to the RSSI\_READBACK register
2. Sets the external LNA signal low (if enabled)
3. Turns off the digital receiver clocks
4. Powers down the synthesizer and the receiver circuitry (ADC, RSSI, IF filter, mixer and LNA)
5. Sets FW\_STATE = PHY\_ON

**CMD\_PHY\_SLEEP (0xBA)**

This command transitions the UHF transceiver to the very low power PHY\_SLEEP state in which the WUC is operational (if enabled) and the BBRAM contents are retained.

It can be issued from the PHY\_OFF or PHY\_ON state.

**CMD\_PHY\_RX (0xB2)**

This command can be issued in the PHY\_ON, PHY\_RX or PHY\_TX states. If the command is issued in the PHY\_ON state, the communications processor performs the following procedure:

1. Powers up the synthesizer
2. Powers up the receiver circuitry (ADC, RSSI, IF filter, mixer and LNA)
3. Sets the RF channel based on the CHANNEL\_FREQ[23:0] setting in BBRAM
4. Sets the synthesizer bandwidth
5. Does the VCO calibration
6. Delays for synthesizer settling

7. Enables the digital receiver blocks
8. Sets external LNA enable signal high (if enabled)
9. Sets FW\_STATE = PHY\_RX

If the command is issued in the PHY\_RX state, the communications processor performs the following procedure:

1. Sets the external LNA signal low(if enabled)
2. Unlocks the AFC and AGC
3. Turns off the receive clocks
4. Sets the RF channel based on the CHANNEL\_FREQ[23:0] setting in BBRAM
5. Sets the synthesizer bandwidth
6. Does the VCO calibration
7. Delays for synthesizer settling
8. Enables the digital receiver blocks
9. Sets external LNA enable signal high (if enabled)
10. Sets FW\_STATE = PHY\_RX

If the command is issued in the PHY\_TX state, the communications processor performs the following procedure:

1. Ramps down the PA
2. Sets external PA signal low (if enabled)
3. Turns off the digital transmit clocks
4. Powers up the receiver circuitry (ADC, RSSI, IF filter, mixer and LNA)
5. Sets the RF channel based on the CHANNEL\_FREQ[23:0] setting in BBRAM
6. Sets the synthesizer bandwidth
7. Does the VCO calibration
8. Delays for synthesizer settling
9. Enables the digital receiver blocks
10. Sets external LNA enable signal high (if enabled)
11. Sets FW\_STATE = PHY\_RX

### **CMD\_PHY\_TX (0xB5)**

This command can be issued in the PHY\_ON, PHY\_TX or PHY\_RX state.

If the command is issued in the PHY\_ON state, the communications processor performs the following procedure:

1. Powers up the synthesizer
2. Sets the RF channel based on the CHANNEL\_FREQ[23:0] setting in BBRAM
3. Sets the synthesizer bandwidth
4. Does the VCO calibration
5. Delays for synthesizer settling
6. Enables the digital transmit blocks
7. Sets the external PA enable signal high (if enabled)
8. Ramps up the PA
9. Sets FW\_STATE = PHY\_TX
10. Transmits data

If the command is issued in the PHY\_TX state, the communications processor performs the following procedure

1. Ramps down the PA
2. Sets the external PA enable signal low (if enabled)
3. Turns off digital transmit clocks
4. Sets the RF channel based on the CHANNEL\_FREQ[23:0] setting in BBRAM
5. Sets the synthesizer bandwidth
6. Does VCO calibration
7. Delays for synthesizer settling
8. Enables the digital transmit blocks
9. Sets the external PA enable signal high (if enabled)
10. Ramps up the PA
11. Sets FW\_STATE = PHY\_TX
12. Transmits data

If the command is issued in the PHY\_RX state, the communications processor performs the following procedure

1. Sets the external LNA signal low(if enabled)
2. Unlocks the AFC and AGC
3. Turns off the receive blocks
4. Powers down the receiver circuitry (ADC, RSSI, IF filter, mixer and LNA)
5. Sets the RF channel
6. Sets the synthesizer bandwidth
7. Delays for synthesizer settling
8. Enables the digital transmit blocks
9. Sets the external PA enable signal high (if enabled)
10. Ramps up the PA
11. Sets FW\_STATE = PHY\_TX
12. Transmits data

### **CMD\_CONFIG\_DEV (0xBB)**

This command interprets the BBRAM contents and configures each of the radio parameters based on these contents. It can be issued from the PHY\_OFF or PHY\_ON state. The only radio parameter that isn't configured on this command is the CHANNEL\_FREQ [23:0] setting, which instead is configured as part of a CMD\_PHY\_TX or CMD\_PHY\_RX command.

The user should write to the entire 64 bytes of the BBRAM and then issue the CMD\_CONFIG\_DEV command, which can be issued in the PHY\_OFF or PHY\_ON states.



**CMD\_GET\_RSSI (0xBC)**

This command turns on the receiver, performs an RSSI measurement on the current channel and then returns to the UHF transceiver to the PHY\_ON state. The command can be issued from the PHY\_ON state. The RSSI result is saved to the RSSI\_READBACK register (Address 0x312). This command can only be issued from the PHY\_ON state.

**CMD\_BB\_CAL (0xBE)**

This command performs an IF filter calibration. It can only be issued in the PHY\_ON state. In many cases it may not be necessary to use this command because an IF filter calibration is automatically performed on the PHY\_OFF to PHY\_ON transition if BB\_CAL = 1 in MODE\_CONTROL register (Address 0x11A).

**CMD\_SYNC (0xA2)**

This command is used to allow the host processor and communications processor to establish communications. . It is required to issue a CMD\_SYNC command during each of the following scenarios:

1. Initialization after application of power
2. On a WUC wake-up
3. Initialization after a CMD\_HW\_RESET
4. After a CMD\_RAM\_LOAD\_DONE command has been issued
5. After issuing a CMD\_SYNC command, the host processor should wait until the CMD\_READY status bit is high. This process ensures that the next command issued by the host processor is processed by the communications processor.

**CMD\_HW\_RESET (0xC8)**

The CMD\_HW\_RESET command performs a full power-down of all hardware and the device enters the PHY\_SLEEP state. This command can be issued in any state and is independent of the state of the communications processor.

**CMD\_RAM\_LOAD\_INIT (0xBF)**

This command prepares the communications processor for a subsequent download of a software module to program RAM. This command should only be issued prior to the program RAM being written to by the host processor.

**CMD\_RAM\_LOAD\_DONE (0xC7)**

This command is only required after a download of a software module to program RAM. It indicates to the communications processor that a software module has been loaded to program RAM. As the program RAM should only be written to in the PHY\_OFF state the CMD\_RAM\_LOAD\_DONE command can only be issued in the PHY\_OFF state. The

command will reset the communications processor and reset the packet RAM. This command should be followed by a CMD\_SYNC command.

### **CMD\_IR\_CAL (0xBD)**

This command performs a fully automatic image rejection calibration on the UHF receiver. The command requires that the IR calibration firmware module has been loaded to the UHF transceiver program RAM. The firmware module is available from Analog Devices.

### **CMD\_AES\_ENCRYPT (0xD0), CMD\_AES\_DECRYPT\_INIT (0xD1) and CMD\_AES\_DECRYPT\_ (0xD2)**

These commands allow AES, 128-bit block encryption and decryption of transmit and receive data using key sizes of 128 bits, 192 bits, or 256 bits.

The AES commands require that the AES firmware module has been loaded to the UHF transceiver program RAM. The AES firmware module is available from Analog Devices.

### **CMD\_RS\_ENCODE\_INIT (0xD1), CMD\_RS\_ENCODE (0xD0), and CMD\_RS\_DECODE (0xD2)**

These commands perform Reed Solomon encoding and decoding of transmit and receive data, thereby allowing detection and correction of errors in the received packet.

These commands require that the Reed Solomon firmware module has been loaded to the UHF transceiver program RAM. The Reed Solomon firmware module is available from Analog Devices.

## **Automatic State Transitions**

On certain events the communications processor can automatically transition the UHF transceiver between states. These automatic transitions are illustrated as dashed lines in **Figure 9** and are explained here.

### **TX\_EOF**

The communications processor will automatically transition the device from the PHY\_TX state to the PHY\_ON state at the end of a packet transmission. On the transition the communications processor performs the following actions:

1. Ramps down the PA
2. Sets the external PA signal low
3. Disables the digital transmitter blocks
4. Powers down the synthesizer
5. Sets FW\_STATE = PHY\_ON

### **RX\_EOF**

The communications processor will automatically transition the device from the PHY\_RX state to the PHY\_ON state at the end of a packet reception. On the transition the communications processor performs the following actions:

1. Copies the measured RSSI to RSSI\_READBACK register
2. Sets the external LNA signal low
3. Disables the digital receiver blocks
4. Powers down the synthesizer and the receiver circuitry (ADC, RSSI, IF filter, mixer and LNA)
5. Sets FW\_STATE = PHY\_ON

### **RX TO TX\_AUTO\_TURNAROUND**

If the RX\_TO\_TX\_AUTO\_TURNAROUND bit in the MODE\_CONTROL register (Address 0x11A) is enabled then the device will automatically transition to the PHY\_TX state at the end of a valid packet reception on the same RF channel frequency. On the transition the communications processor performs the following actions:

1. Sets the external LNA signal low
2. Unlocks the AGC and AFC (if enabled)
3. Disables the digital receiver blocks
4. Powers down the receiver circuitry (ADC, RSSI, IF filter, mixer and LNA)
5. Sets RF channel frequency (same as previous receive channel frequency)
6. Sets the synthesizer bandwidth
7. Does VCO calibration
8. Delays for synthesizer settling
9. Enable the digital transmitter blocks
10. Sets the external PA signal high ( if enabled)
11. Ramps up the PA
12. Sets FW\_STATE= PHY\_TX
13. Transmits data

In sport mode the RX\_TO\_TX\_AUTO\_TURNAROUND transition is disabled.

### **TX\_TO\_RX\_AUTO\_TURNAROUND**

If the TX\_TO\_RX\_AUTO\_TURNAROUND bit in the MODE\_CONTROL register (Address 0x11A) is enabled then the device will automatically transition to PHY\_RX state at the end of a packet transmission, on the same RF channel frequency. On the transition the communications processor performs the following actions:

1. Ramps down the PA
2. Sets the external PA signal low
3. Disables the digital transmitter blocks
4. Powers up the receiver circuitry (ADC, RSSI, IF filter, mixer and LNA)
5. Sets RF channel frequency (same as previous transmit channel frequency)
6. Sets the synthesizer bandwidth

7. Does VCO calibration
8. Delays for synthesizer settling
9. Turns on AGC and AFC (if enabled)
10. Enables the digital receiver blocks
11. Sets the external LNA signal high (if enabled)
12. Sets FW\_STATE= PHY\_RX

In sport mode the TX\_TO\_RX\_AUTO\_TURNAROUND transition is disabled.

## WUC timeout

The UHF transceiver can use the WUC to wake from sleep on a timeout of the hardware timer. The device wakes into the PHY\_OFF state.

## State Transition and Command Timing

The execution times for all radio state transitions are detailed in Table 83 and Table 84. Note that these times are typical and can vary depending on the BBRAM configuration.

**Table 83: UHF Transceiver Command Execution Times and State Transition Times That Are Not Related to PHY\_TX or PHY\_RX**

Command/Bit	Command Initiated By	Present State	Next State	Transition Time (μs), Typical	Condition
CMD_HW_RESET	Host	Any	PHY_SLEEP	1	
CMD_PHY_SLEEP	Host	<b>PHY_OFF</b>	PHY_SLEEP	22.3	
CMD_PHY_SLEEP	Host	<b>PHY_ON</b>	PHY_SLEEP	24.1	
CMD_PHY_OFF	Host	<b>PHY_ON</b>	PHY_OFF	19	
CMD_PHY_ON	Host	<b>PHY_OFF</b>	PHY_ON	248	Including IF filter calibration
CMD_GET_RSSI	Host	<b>PHY_ON</b>	PHY_ON	612.5	
CMD_CONFIG_DEV	Host	<b>PHY_OFF</b>	PHY_OFF	66.8	
CMD_CONFIG_DEV	Host	<b>PHY_ON</b>	PHY_ON	66.8	
CMD_BB_CAL	Host	<b>PHY_ON</b>	PHY_ON	211	
Wake-Up from PHY_SLEEP, (WUC Timeout)	Automatic	<b>PHY_SLEEP</b>	PHY_OFF	310 + 252.8	The 310 μs is for startup of the 26 MHz crystal (7 pF load capacitance, T <sub>A</sub> = 25°C).
Wake-Up from PHY_SLEEP, (CS Low)	Host	<b>PHY_SLEEP</b>	PHY_OFF	310 + 252.8	The 310 μs is for startup of the 26 MHz crystal (7 pF load capacitance, T <sub>A</sub> = 25°C)
Cold Start	Application of power	<b>N/A</b>	PHY_OFF	310 + 252.8	The 310 μs is for startup of the 26 MHz crystal (7 pF load capacitance, T <sub>A</sub> = 25°C)

**Table 84: UHF Transceiver State Transition Times Related to PHY\_TX and PHY\_RX**

Mode	Command/Bit/ Automatic Transition	Present State	Next State	Transition Time (μs), Typical <sup>1,2</sup>	Condition
Packet	CMD_PHY_ON	PHY_TX	PHY_ON	$T_{EOP} + 10.8 + T_{PARAM\_DOWN} + 36$	
Packet	CMD_PHY_ON	PHY_RX	PHY_ON	$5 + T_{BYTE} + 38.2$	CMD_PHY_ON issued during search for preamble
				41.7	CMD_PHY_ON issued during preamble qualification
				41.2	CMD_PHY_ON issued during sync word qualification
				$T_{EOP} + 31.7$	CMD_PHY_ON issued during Rx data (after a sync word)
Packet	CMD_PHY_TX	PHY_ON	PHY_TX	$293 + T_{PARAM\_UP} + 3$	
Packet	CMD_PHY_TX	PHY_RX	PHY_TX	$5 + T_{BYTE} + 305.3 + T_{PARAM\_UP} + 3$	CMD_PHY_TX issued during search for preamble
				$308.5 + T_{PARAM\_UP} + 3$	CMD_PHY_TX issued during preamble qualification
				$308 + T_{PARAM\_UP} + 3$	CMD_PHY_TX issued during sync word qualification
				$298.5 + T_{EOP} + T_{PARAM\_UP} + 3$	CMD_PHY_TX issued during Rx data (after a sync word)
Packet	CMD_PHY_TX	PHY_TX	PHY_TX	$T_{EOP} + 10.8 + T_{PARAM\_DOWN} + 297 + T_{PARAM\_UP} + 3$	CMD_PHY_TX issued during packet transmission
Packet	RX_TO_TX_ AUTO_TURNAROUND	PHY_RX	PHY_TX	$287.3 + T_{PARAM\_UP} + 3$	
Packet	CMD_PHY_RX	PHY_ON	PHY_RX	300	
Packet	CMD_PHY_RX	PHY_TX	PHY_RX	$T_{EOP} + 10.8 + T_{PARAM\_DOWN} + 317$	CMD_PHY_RX issued during packet transmission
Packet	CMD_PHY_RX	PHY_RX	PHY_RX	$5 + T_{BYTE} + 305.2$	CMD_PHY_RX issued during search for preamble
				308.4	CMD_PHY_RX issued during preamble qualification
				307.9	CMD_PHY_RX issued during sync word qualification
				$T_{EOP} + 298.4$	CMD_PHY_RX issued during Rx data (after a sync word)

#### UHF Transceiver State Transition Times Related to PHY\_TX and PHY\_RX continued

$$^1 T_{PARAM\_UP} = T_{PARAM\_DOWN} = PA\_LEVEL\_MCR / \{2^{(9-PA\_RAMP)} \times DATA\_RATE \times 100\}$$

Where PA\_LEVEL\_MCR sets the maximum PA output power (PA\_LEVEL\_MCR register, Address 0x307),

PA\_RAMP sets the PA ramp rate (RADIO\_CFG\_8 register, Address 0x114), and

DATA\_RATE sets the transmit data rate (RADIO\_CFG\_0 register, Address 0x10C and RADIO\_CFG\_1 register, Address 0x10D).

$$^2 T_{BYTE} = \text{one byte period } (\mu s), T_{EOP} = \text{time to end of packet } (\mu s).$$

Mode	Command/Bit/ Automatic Transition	Present State	Next State	Transition Time ( $\mu$ s), Typical <sup>1,2</sup>	Condition
Packet	TX_TO_RX_ AUTO_TURNAR ROUND	PHY_TX	PHY_RX	$10.8 + T_{\text{PARAMP\_DOWN}} + 306$	
Packet	TX_EOF	PHY_TX	PHY_ON	$10.8 + T_{\text{PARAMP\_DOWN}} + 36$	
Packet	RX_EOF	PHY_RX	PHY_ON	17.2	
Sport	CMD_PHY_ON	PHY_TX	PHY_ON	$10.8 + T_{\text{PARAMP\_DOWN}} + 36$	
Sport	CMD_PHY_ON	PHY_RX	PHY_ON	$5 + T_{\text{BYTE}} + 38.2$	CMD_PHY_ON issued during search for a preamble
				41.7	CMD_PHY_ON issued during preamble qualification
				41.2	CMD_PHY_ON issued during sync word qualification
				31.7	CMD_PHY_ON issued during Rx data (after a sync word)
Sport	CMD_PHY_TX	PHY_ON	PHY_TX	$293 + T_{\text{PARAMP\_UP}} + 3$	
Sport	CMD_PHY_TX	PHY_RX	PHY_TX	$5 + T_{\text{BYTE}} + 305.3 + T_{\text{PARAMP\_UP}} + 3$	CMD_PHY_TX issued during search for a preamble
				$308.5 + T_{\text{PARAMP\_UP}} + 3$	CMD_PHY_TX issued during preamble qualification
				$308 + T_{\text{PARAMP\_UP}} + 3$	CMD_PHY_TX issued during sync word qualification
				$298.5 + T_{\text{PARAMP\_UP}} + 3$	CMD_PHY_TX issued during Rx data (after a sync word)
Sport	CMD_PHY_TX	PHY_TX	PHY_TX	$10.8 + T_{\text{PARAMP\_DOWN}} + 297 + T_{\text{PARAMP\_UP}} + 3$	
Sport	RX_TO_TX_ AUTO_TURNAR ROUND	PHY_RX	PHY_TX	$287.3 + T_{\text{PARAMP\_UP}} + 3$	
Sport	CMD_PHY_RX	PHY_ON	PHY_RX	300	
Sport	CMD_PHY_RX	PHY_TX	PHY_RX	$+ T_{\text{PARAMP\_DOWN}} + 317$	
Sport	CMD_PHY_RX	PHY_RX	PHY_RX	$5 + T_{\text{BYTE}} + 305.2$	CMD_PHY_RX issued during search for a preamble
				308.4	CMD_PHY_RX issued during preamble qualification
				307.9	CMD_PHY_RX issued during sync word qualification
				298.4	CMD_PHY_RX issued during Rx data (after a sync word)
Sport	TX_TO_RX_AU TO_TURNARO UND	PHY_TX	PHY_RX	$10.8 + T_{\text{PARAMP\_DOWN}} + 306$	

<sup>1</sup>  $T_{\text{PARAMP\_UP}} = T_{\text{PARAMP\_DOWN}} = \text{PA\_LEVEL\_MCR} / \{2^{(9-\text{PA\_RAMP})} \times \text{DATA\_RATE} \times 100\}$

Where PA\_LEVEL\_MCR sets the maximum PA output power (PA\_LEVEL\_MCR register, Address 0x307),

PA\_RAMP sets the PA ramp rate (RADIO\_CFG\_8 register, Address 0x114), and

DATA\_RATE sets the transmit data rate (RADIO\_CFG\_0 register, Address 0x10C and RADIO\_CFG\_1 register, Address 0x10D).

<sup>2</sup>  $T_{\text{BYTE}} = \text{one byte period } (\mu\text{s}), T_{\text{EOP}} = \text{time to end of packet } (\mu\text{s}).$

## Packet Mode

The on-chip communications processor can be configured for use with a wide variety of packet based radio protocols using 2FSK/MSK /GFSK/GMSK /OOK modulation. The general packet format, when using the packet management features of the communications processor are described in Table 85. To use the packet management features the DATA\_MODE setting should be set to packet mode in the PACKET\_LENGTH\_CONTROL register (Address 0x126). 240 bytes of dedicated packet RAM is available to store, transmit and receive packets.

In transmit mode preamble, sync word and CRC can be added by the communications processor to the data stored in the packet RAM for transmission. In addition, all packet data after the sync word can be optionally whitened, Manchester encoded or 8b/10b encoded on transmission and decoded on reception.

In receive mode the communications processor can be used to qualify received packets based on the preamble detection, sync word detection, CRC detection or address match and generate an interrupt on the IRQ\_GP3 pin. On reception of valid packet, the received payload data is loaded to packet RAM memory.

**Table 85: Packet Structure Overview**

Packet Format Options	Packet Structure						
	Preamble	Sync	Payload			CRC	Postamble
			Length	Address	Payload Data		
Field Length	1 byte to 256 bytes	1 bit to 24 bits	1 byte	1 byte to 9 bytes	0 bytes to 240 bytes	2 bytes	2 bytes
Optional Field in Packet Structure	X <sup>1</sup>	X	Yes <sup>2</sup>	Yes	Yes	Yes	X
Comms Processor Adds in Tx, Removes in Rx	Yes	Yes	X	X	X	Yes	Yes
Host Writes These Fields to Packet RAM	X	X	Yes	Yes	Yes	X	X
Whitening/Dewhitening	X	X	Yes	Yes	Yes	Yes	X
Manchester Encoding/Decoding	X	X	Yes	Yes	Yes	Yes	X
8b/10b Encoding/ Decoding	X	X	Yes	Yes	Yes	Yes	X
Configurable Parameter	Yes	Yes	Yes	Yes	Yes	Yes	X
Receive Interrupt on Valid Field Detection	Yes	Yes	X	Yes	X	Yes	X
Programmable Field Error Tolerance	Yes	Yes	X	X	X	X	X
Programmable Field Offset	X	X	X	Yes	X	X	X

<sup>1</sup> X indicates that the packet format option is not supported

<sup>2</sup> Yes indicates that the packet format option is supported

## Preamble

This is a mandatory part of the packet that is automatically added by the communications processor when transmitting a packet and removed after receiving a packet. The preamble is a 0x55 sequence, with a programmable length between 1 byte and 256 bytes which is set in `PREAMBLE_LEN` register (Address 0x11D). It is necessary to have preamble at the beginning of the packet to allow time for the receiver AGC, AFC and clock and data recovery circuitry to settle before the start of the sync word. The required preamble length depends on the radio configuration.

In receive mode the UHF Transceiver can use a preamble qualification circuit to detect preamble and interrupt the host processor. The preamble qualification circuit tracks the received frame as a sliding window. The window is three bytes in length, and the preamble pattern is fixed at 0x55. The preamble bits are examined in “01” pairs. If either bit or both bits are in error, the pair is deemed erroneous. The possible erroneous pairs are “00”, “11” and “10”. The number of erroneous pairs tolerated in the preamble can be set using the `PREAMBLE_MATCH` register value (Address 0x11B) according to Table 86.

**Table 86: Preamble Detection Tolerance (`PREAMBLE_MATCH`, Address 0x11B)**

Value	Description
0x0C	No errors allowed.
0x0B	One erroneous bit-pair allowed in 12 bit-pairs.
0x0A	Two erroneous bit-pairs allowed in 12 bit-pairs.
0x09	Three erroneous bit-pairs allowed in 12 bit-pairs.
0x08	Four erroneous bit-pairs allowed in 12 bit-pairs.
0x00	Preamble detection disabled.

If `PREAMBLE_MATCH` is set to 0x0C the UHF Transceiver must receive 12 consecutive “01” pairs (3 bytes) to confirm that a valid preamble has been detected. The user can select the option to automatically lock the AFC and/or AGC once the qualified preamble is detected.

The AFC lock on preamble detection can be enabled by setting `AFC_LOCK_MODE` = 0x3 in the `RADIO_CFG_10` register (Address 0x116).

The AGC lock on preamble detection can be enabled by setting `AGC_LOCK_MODE` = 0x3 in the `RADIO_CFG_7` register (Address 0x113).

After a preamble is detected and the end of the preamble has been reached, the communications processor will search for sync word. The search for the sync word last for a duration equal to the sum of the number of programmed sync word bits, plus the preamble matching tolerance (in bits) plus 16 bits. This is illustrated in Figure 10.

If the sync word routine is detected during this duration the communications processor loads the received payload to packet RAM and compute the CRC (if enabled). If sync word is not detected during this duration the communications processor continues searching for the preamble.

Preamble detection can be disabled by setting the `PREAMBLE_MATCH` register to 0x00.

To enable an interrupt upon preamble detection, the user must set `INTERRUPT_PREAMBLE_DETECT` = 1 in the `INTERRUPT_MASK_0` (Address 0x100).



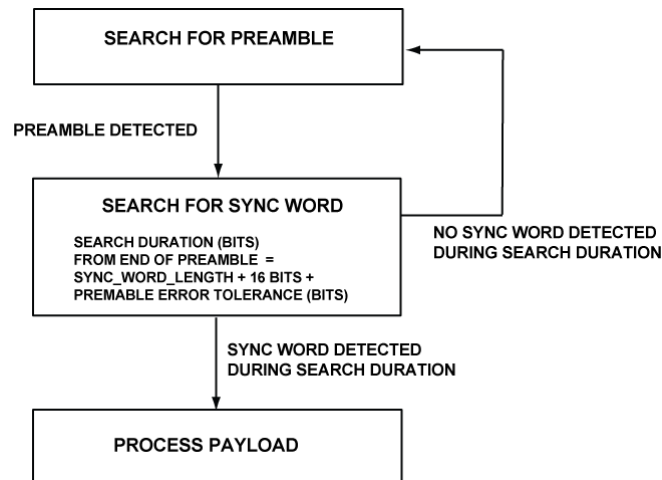


Figure 10: Search for Preamble and Sync Word Routine

### Sync Word

Sync word is the synchronization word and is used by the receiver for byte level synchronization, while also providing an optional interrupt on detection. It is automatically added to the packet by the communications processor in transmit mode and removed during reception of a packet.

The value of the sync word is set in the SYNC\_BYTE\_0, SYNC\_BYTE\_1 and SYNC\_BYTE\_2 registers (Address 0x121, 0x122 and 0x123 respectively). The sync word is transmitted most significant bit first starting with SYNC\_BYTE\_0. The sync word matching length at the receiver is set using SYNC\_WORD\_LENGTH in the SYNC\_CONTROL register (Address 0x120) and can be 1 bit to 24 bits in length. The transmitted sync word is a multiple of 8 bits. Hence, for non-byte length sync words, the transmitted sync pattern should be appended with the preamble pattern as described in Figure 11 and Table 87.

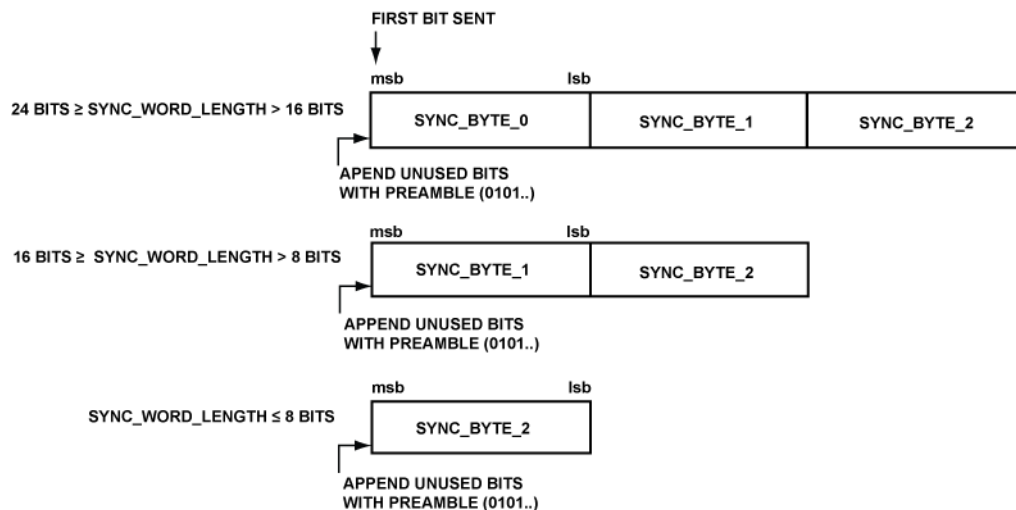


Figure 11: Transmit Sync Word Configuration

Table 87: Sync Word Programming Examples

Required sync word (binary, first bit being first in time)	sync_word_length bits in SYNC_CONTROL register (Address 0x120)	sync_byte_0	sync_byte_1	sync_byte_2	Transmitted sync word (binary, first bit being first in time)	Receiver sync word match length (bits)
0001 0010 0011 0100 0101 0110	24	0x12	0x34	0x56	0001_0010_0011_0100_0101_0110	24
1110 1001 1100 1010 0010 0	21	0x5D	0x39	0x44	0101_1101_0011_1001_0100_0100	21
0001 0010 0011 0100	16	0xXX	0x12	0x34	0001_0010_0011_0100	16
0111 0000 1110	12	0xXX	0x57	0x0E	0101_0111_0000_1110	12
0001 0010	8	0xXX	0xXX	0x12	0001_0010	8
0111 00	6	0xXX	0xXX	0x5C	0101_1100	6

In receive mode the UHF Transceiver can provide an interrupt on reception of the sync word sequence programmed in the SYNC\_BYTE\_0, SYNC\_BYTE\_1 and SYNC\_BYTE\_2 registers. This feature can be used to alert the host processor that a qualified sync word has been received. An error tolerance parameter can also be programmed that accepts a valid match when up to three bits of the sync word sequence are incorrect. The error tolerance value is set using the SYNC\_ERROR\_TOL setting in the SYNC\_CONTROL register (Address 0x120:) as described in Table 88.

Table 88: Sync Word Detection Tolerance

Value	Description
00	No bit errors allowed
01	1 bit error allowed
10	2 bit errors allowed
11	3 bit errors allowed

## Choice of Sync Word

The sync word should be chosen to have low correlation with the preamble and have good autocorrelation properties. When the AFC is set to lock on detection of sync word (AFC\_LOCK\_MODE = 3 and PREAMBLE\_MATCH = 0), the sync word should be chosen to be dc free, and it should have a run length limit not greater than four bits

## Payload

The host processor writes the transmit data payload to the packet RAM. The location of transmit data in the packet RAM is defined by the TX\_BASE\_ADR value register (Address 0x124). The TX\_BASE\_ADR value is the location of the first byte of the transmit payload data in packet RAM. On reception of a valid sync word the communications processor automatically loads the receive payload to the packet RAM. The RX\_BASE\_ADR value (Address 0x125) sets the location in packet RAM of the first byte of the received payload.

## Byte Orientation

The over-the-air arrangement of each transmitted packet RAM byte can be set to MSB first or LSB first using the DATA\_BYTE setting in the PACKET\_LENGTH\_CONTROL register (Address 0x126:). The same orientation setting should be used on the transmit and receive side of the RF link.

## Packet Length Modes

The UHF Transceiver can be used in both fixed and variable length packet systems. Fixed or variable length packet mode is set using the PACKET\_LEN\_VARIABLE in the PACKET\_LENGTH\_CONTROL register (Address 0x126:).

For a fixed packet length system the length of the transmit and received payload is set by the PACKET\_LENGTH\_MAX register (Address 0x127). The payload length is defined as the number of bytes from the end of sync word to the start of the CRC.

In variable packet length mode the communications processor extracts the length field from the received payload data. In transmit mode, the length field must be the first byte in the transmit payload.

The communications processor calculates the actual received payload length as:

$$\text{Rx payload Length} = \text{Length} + \text{LENGTH\_OFFSET} - 4$$

Where:

Length is the length field (first byte in received payload)

LENGTH\_OFFSET (:) is a programmable offset set in the PACKET\_LENGTH\_CONTROL register (Address 0x126)

The LENGTH\_OFFSET value allows compatibility with systems where the length field in the proprietary packet may also include the length of the CRC and/or the sync word. The UHF Transceiver defines the payload length as the number of bytes from the end of sync

word to the start of the CRC. In variable packet length mode the PACKET\_LENGTH\_MAX value defines the maximum packet length that can be received. This is described in Figure 12.

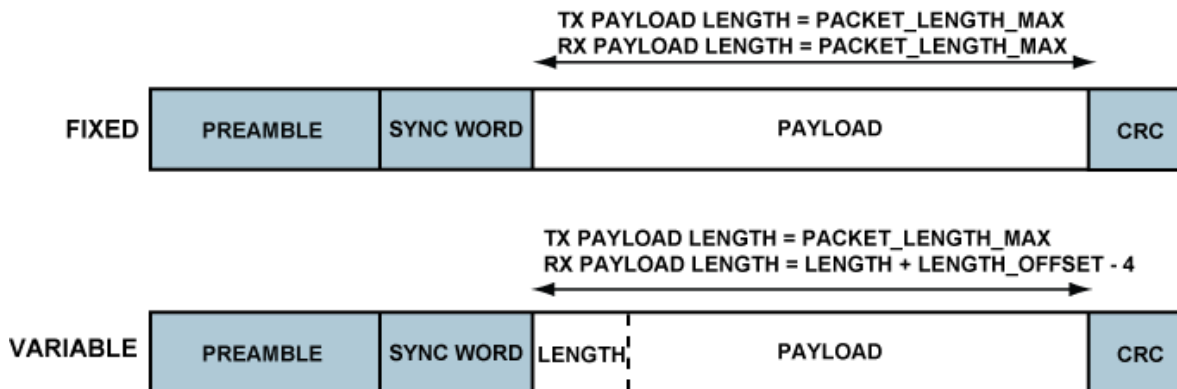


Figure 12: Payload Length in Fixed and Variable Length Packet Modes

## Addressing

The UHF Transceiver provides a very flexible address matching scheme allowing matching of a single address, multiple addresses and broadcast addresses. The address information can be included at any section of the transmit payload. The location of the starting byte of the address data in the received payload is set in the ADDRESS\_MATCH\_OFFSET register (Address 0x129) as illustrated in Figure 13. The number of bytes in the first address field is set in the ADDRESS\_LENGTH register (Address 0x12A). These settings allow the communications processor to extract the address information from the received packet. The address data is then compared against a list of known addresses which are stored in BBRAM (Address 0x12B to address 0x13D). Each stored address byte has an associated mask byte thereby allowing matching of partial sections of the address bytes, which is useful for checking broadcast addresses or a family of addresses that have a unique identifier in the address sequence. The format and placement of the address information in the payload data should match the address check settings at the receiver to ensure exact address detection and qualification. Table 89 shows the register locations in the BBRAM that are used for setup of the address checking. When Register 0x12A (number of bytes in the first address field) is set to 0x00, address checking is disabled.

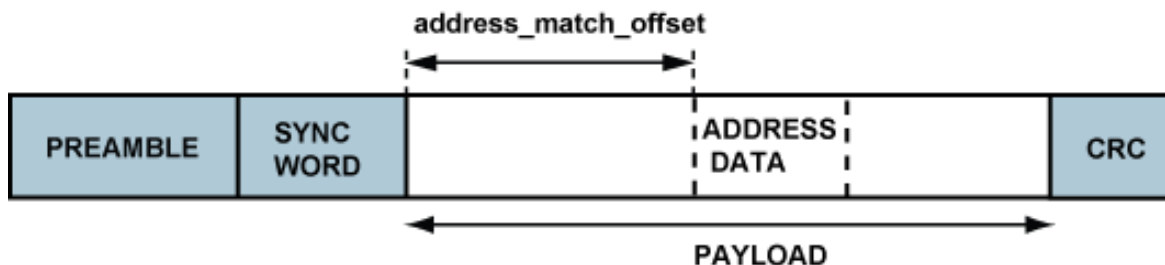


Figure 13: Address Match Offset

**Table 89: Address Check Register Setup**

Address (BBRAM)	Description
0x129, ADDRESS_MATCH_OFFSET	Position of first address byte in packet ( First byte after sync word = 0)
0x12A, ADDRESS_LENGTH	Number of Bytes in Address field ( $N_{ADR\_1}$ )
0x12B	Address Match Byte 0
0x12C	Address Mask Byte 0
0x12D	Address Match Byte 1
0x12E	Address Mask Byte 1
:	:
	Address Match Byte $N_{ADR\_1}-1$
	Address Mask Byte $N_{ADR\_1}-1$
	0x00 to end or $N_{ADR\_2}$ for another address check sequence

The host processor should set the INTERRUPT\_ADDRESS\_MATCH bit in the INTERRUPT\_SOURCE\_0 register (Address 0x336) if an interrupt is required on the IRQ\_GP3 pin.

#### Example Address Check

Consider a system with 32-bit address lengths, in which the first byte is located in the 10<sup>th</sup> byte of the received payload data. The system also uses broadcast addresses in which the first byte is always 0xAA. To match the exact address, 0xABCDEF01 OR any broadcast address of the form 0xAAXXXXXX the UHF transceiver must be configured as shown in .

Table 90: Example Address Check Configuration

BBRAM Address (Hex)	Value (Hex)	Description
0x129	0x09	Location in payload of 1 <sup>st</sup> address byte
0x12A	0x04	Number of bytes in the 1 <sup>st</sup> address field, $N_{ADR\_1}=4$
0x12B	0xAB	Address Match Byte 0
0x12C	0xFF	Address Mask Byte 0
0x12D	0xCD	Address Match Byte 1
0x12E	0xFF	Address Mask Byte 1
0x12F	0xEF	Address Match Byte 2
0x130	0xFF	Address Mask Byte 2
0x131	0x01	Address Match Byte 3
0x132	0xFF	Address Mask Byte 3
0x133	0x04	Number of bytes in the 2nd address field, $N_{ADR\_2}=4$
0x134	0xAA	Address Match Byte 0
0x135	0xFF	Address Mask Byte 0
0x136	0x00	Address Match Byte 1
0x137	0x00	Address Mask Byte 1
0x138	0x00	Address Match Byte 2
0x139	0x00	Address Mask Byte 2
0x13A	0x00	Address Match Byte 3
0x13B	0x00	Address Mask Byte 3
0x13C	0x00	End of addresses (indicated by 0x00)
0x13D	0xFF	Don't Care

## CRC

An optional CRC-16 can be appended to the packet by setting  $CRC\_EN = 1$  in the  $PACKET\_LENGTH\_CONTROL$  register (Address 0x126:). In receive mode, this bit enable CRC detection on the receive packet.

The default polynomial is used if  $PROG\_CRC\_EN = 0$  in the Symbol mode register (Address 0x11C:). The default CRC polynomial is

$$g(x) = x^{16} + x^{12} + x^5 + 1$$

Any other 16-bit polynomial can be used if  $PROG\_CRC\_EN = 1$ , and the polynomial is set in  $CRC\_POLY\_0$  and  $CRC\_POLY\_1$  (Address 0x11E and 0x11F). The setup of the CRC is described in Table 91.

**Table 91: CRC Setup (prog\_crc\_en = 1)**

crc_en (0x126)	prog_crc_en (0x11C)	Description
0	0	CRC disabled in transmit and CRC detection disabled in receive
0	1	CRC disabled in transmit and CRC detection disabled in receive
1	0	CRC enabled in transmit and CRC detection enabled in receive, with default CRC polynomial
1	1	CRC enabled in transmit and CRC detection enabled in receive, with CRC polynomial defined by CRC_POLY_0 and CRC_POLY_1

To convert a user defined polynomial to a two byte value, the polynomial should be first written in binary format, with the  $x^{16}$  bit discarded as it is assumed equal to 1. The remaining 16 bits then make up the CRC\_POLY\_0 (MSB) and CRC\_POLY\_1 (LSB) values. An example of setting the 16 bit CRCs are shown in Table 92.

**Table 92: Example: Programming of crc\_poly\_0 and crc\_poly\_1**

Polynomial	Binary Format	crc_poly_0	crc_poly_1
$X^{16} + X^{15} + X^2 + 1$ (CRC – 16- IBM)	1_1000_0000_0000_0101	0x80	0x05
$X^{16} + X^{13} + X^{12} + X^{11} + X^{10} + X^8 + X^6 + X^5 + X^2 + 1$ (CRC – 16 – DNP)	1_0011_1101_0110_0101	0x3D	0x65

To enable CRC detection on the receiver, with the default CRC or user defined 16-bit CRC the CRC\_EN in the PACKET\_LENGTH\_CONTROL register (Address 0x126:) should be set to 1. An interrupt can be generated on reception of CRC verified packet.

### Post-amble

The communications processor automatically appends 2 bytes of post-amble to the end of the transmitted packet. Each byte of post-amble is 0x55. The first byte is transmitted immediately after the CRC. The PA ramp down begins immediately after the first post-amble byte. The second byte is transmitted while the PA is ramping down.

On the receiver, if the received packet is valid the RSSI is automatically measured during the first post-amble byte and the result stored in the RSSI\_READBACK register (Address 0x312). The RSSI is measured by the communications processor 17us after the last CRC bit.

## Transmit Packet Timing

The PA ramp timing in relation to the transmit packet data is described in Figure 14. After the CMD\_PHY\_TX command has been issued a VCO calibration is carried out followed by a delay for synthesizer settling. The PA ramp follows the synthesizer settling. After the PA has ramped up to the programmed rate there is 1 byte delay before the start of modulation (preamble). At the beginning of the second byte of post-amble the PA ramps down. The communications processor then transitions to the PHY\_ON state or PHY\_RX state (if the TX\_AUTO\_TURNAROUND bit is enabled or CMD\_PHY\_RX command is issued).

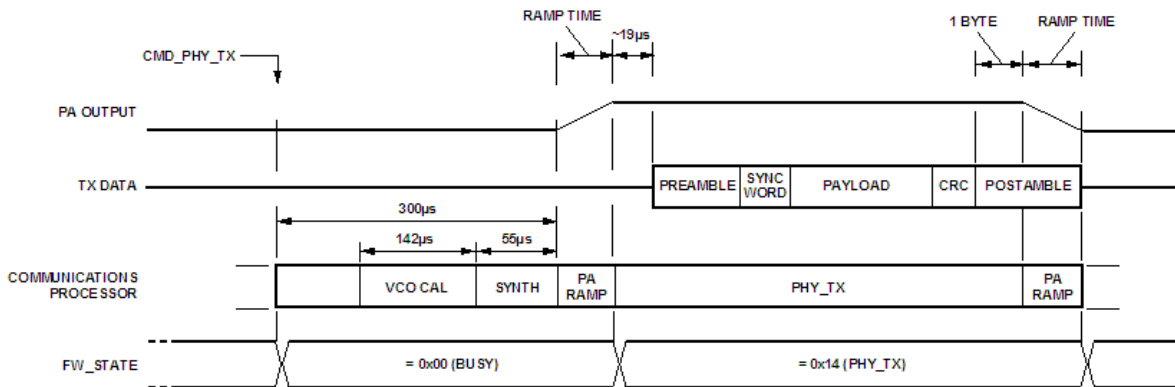


Figure 14: Transmit Packet Timing

## Data Whitening

Data whitening can be employed to avoid long runs of 1's or 0's in the transmitted data stream. This ensures sufficient bit transitions in the packet, which aids in receiver clock and data recovery because the encoding breaks up long runs of 1's or 0's in the transmit packet. The data, excluding preamble and sync word, is automatically whitened before transmission by XOR-ing the data with an 8-bit pseudo-random sequence. , thereby reversing the whitening. The linear feedback shift register polynomial used is  $X^7 + X^1 + 1$ . Data whitening and de-whitening is enabled by setting DATA\_WHITENING = 1 in the SYMBOL\_MODE register (Address 0x11C:).

## Manchester Encoding

Manchester encoding can be used to ensure a DC-free (zero mean) transmission. The encoded over-the-air bit rate (chip rate) is double the rate set by the DATA\_RATE variable (Address 0x10C and Address 0x10D). A binary zero is mapped to "10" and a binary one is mapped to "01". The Manchester encoding and decoding are applied to the payload data and the CRC. It is recommended to used Manchester Encoding for OOK modulation. Manchester encoding and decoding is enabled by setting MANCHESTER\_ENC =1 in the SYMBOL\_MODE register (Address 0x11C:).



## 8b/10b Encoding

8b/10b encoding is a byte orientated encoding scheme that maps an 8-bit byte to a 10-bit data block. It ensures the maximum number of consecutive 1's or 0's (that is run length) in any 10 bit transmitted symbol is 5. The advantage of this encoding scheme is that DC balancing is employed without the efficiency loss of Manchester encoding. The rate loss for Manchester encoding is 0.5 while for 8b/10b encoding the rate loss is 0.8. Encoding and decoding are applied to the payload data and the CRC. The 8b/10b encoding and decoding is enabled by setting EIGHT\_TEN\_ENC =1 in the SYMBOL\_MODE register (Address 0x11C

## Sport Mode

In applications where the UHF transceiver communications processor cannot accommodate the packet format (e.g. payload length > 240 bytes is required), it is possible to bypass its packet handling features and use the SPORT interface for transmit and receive data.

The SPORT interface is a high speed synchronous serial interface. The SPORT interface allows interfacing to processors and DSPs without glue logic.

SPORT mode is enabled using the DATA\_MODE setting in the PACKET\_LENGTH\_CONTROL register, (Address 0x126) as described in Table 93. The SPORT mode interface is on the GPIO pins (GP0, GP1, GP2, GP4 and GP5). These GPIO pins can be configured using the GPIO\_CONFIGURE setting (Address 0x3FA) as described in Table 94 .

SPORT mode provides a receive interrupt source on GP4. This interrupt source can be configured to provide an interrupt, or strobe signal, on either preamble detection or sync word detection. The type of interrupt is configured using the GPIO\_CONFIGURE setting.

## Packet Structure in SPORT MODE

In SPORT mode the host processor has full control over the packet structure. However, the preamble frame is still required to allow sufficient bits for receiver settling (AGC, AFC and CDR). In SPORT mode sync word detection is not mandatory in the UHF transceiver, but can be enabled to provide byte level synchronization for the host processor via the sync word detect interrupt or strobe on GP4. The general format of a SPORT mode packet is shown in Figure 15.



Figure 15: General SPORT Mode Packet

## SPORT Mode in Transmit

Figure 16 illustrates the operation of the SPORT interface in the transmit case. Once in the PHY\_TX state and with SPORT mode enabled, the data input of the transmitter is fully controlled by the SPORT interface (pin GP1). The transmit clock appears on GP2 pin. The transmit data from the host processor should be synch-ronized with this clock. The FW\_STATE variable in the status word or the CMD\_FINISHED interrupt can be used to indicate when the UHF Transceiver has reached the PHY\_TX state and, there-fore, is ready to begin transmitting data. The UHF Transceiver keeps transmitting the serial data presented at the GP1 input until the host processor issues a command to exit the PHY\_TX state.

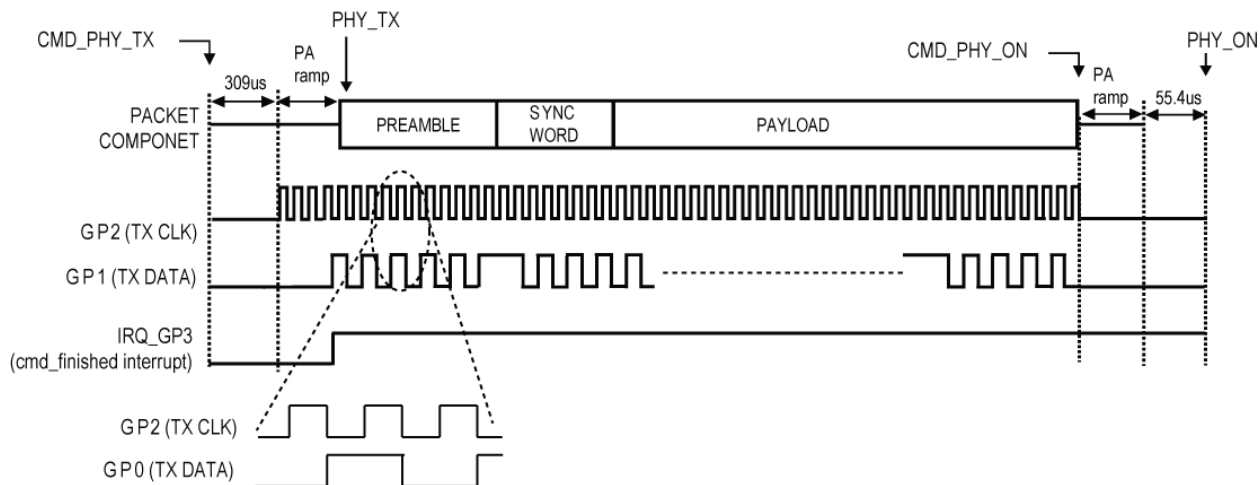


Figure 16: SPORT Mode Operation in Transmit Mode

## SPORT Mode in Receive

The SPORT interface supports the receive operation with a number of modes to suit particular signalling requirements. The receive data appears on the GP0 pin whereas the receive synchronized clock appears on the GP1 pin. The GP4 pin provides an interrupt or strobe signal on either preamble or sync word detection. Once enabled, the interrupt signal and strobe signals remain operational while in the PHY\_RX state. The strobe signal gives a single high pulse of 1-bit duration every eight bits. The strobe signal is most useful when used with sync word detection because it is synchronized to the sync word and strobes the first bit in every byte.

**Table 93: SPORT Mode Setup**

data_mode (Address 0x126)	Description	GPIO Configuration
data_mode = 0	Packet Mode enabled. Packet handling controlled by communications processor.	-
data_mode = 1	SPORT mode enabled. Rx data and Rx clock enabled in PHY_RX (gpio_configure = 0xA0, 0xA3, 0xA6)  Rx clock enabled in PHY_RX and Rx data enabled on preamble detect (gpio_configure = 0xA1, 0xA2, 0xA4, 0xA5, 0xA7, 0xA8)	GP0: Rx Data GP1: Tx Data GP2: Tx/Rx Clock GP4: interrupt or strobe enabled on preamble detect (depends on gpio_configure) GP5: depends on gpio_configure
data_mode = 2	SPORT mode enabled.  Rx data and Rx clock enabled in PHY_RX if gpio_configure = 0xA0, 0xA3, 0xA6  Rx clock enabled in PHY_RX and Rx data enabled on preamble detect if gpio_configure = 0xA1, 0xA2, 0xA4, 0xA5, 0xA7, 0xA8	GP0: Rx Data GP1: Tx Data GP2: Tx/Rx Clock GP4: interrupt or strobe enabled on sync word detect (depends on gpio_configure) GP5: depends on gpio_configure

**Table 94: GPIO Functionality in Sport Mode**

GPIO_CONFIGURE	GP0	GP1	GP2	GP4	XOSC32KP_GP5_ATB1
0xA0	Rx data	Tx data	Tx/Rx clock	Not used	Not used
0xA1	Rx data	Tx data	Tx/Rx clock	Interrupt	Not used
0xA2	Rx data	Tx data	Tx/Rx clock	Strobe	Not used
0xA3	Rx data	Tx data	Tx/Rx clock	Not used	32.768 kHz XTAL input
0xA4	Rx data	Tx data	Tx/Rx clock	Interrupt	32.768 kHz XTAL input
0xA5	Rx data	Tx data	Tx/Rx clock	Strobe	32.768 kHz XTAL input
0xA6	Rx data	Tx data	Tx/Rx clock	Not used	EXT_UC_CLK output
0xA7	Rx data	Tx data	Tx/Rx clock	Interrupt	EXT_UC_CLK output
0xA8	Rx data	Tx data	Tx/Rx clock	Strobe	EXT_UC_CLK output

## GPIO Functionality in Sport Mode

### Mode 0 (GPIO\_CONTROL = 0xA0)

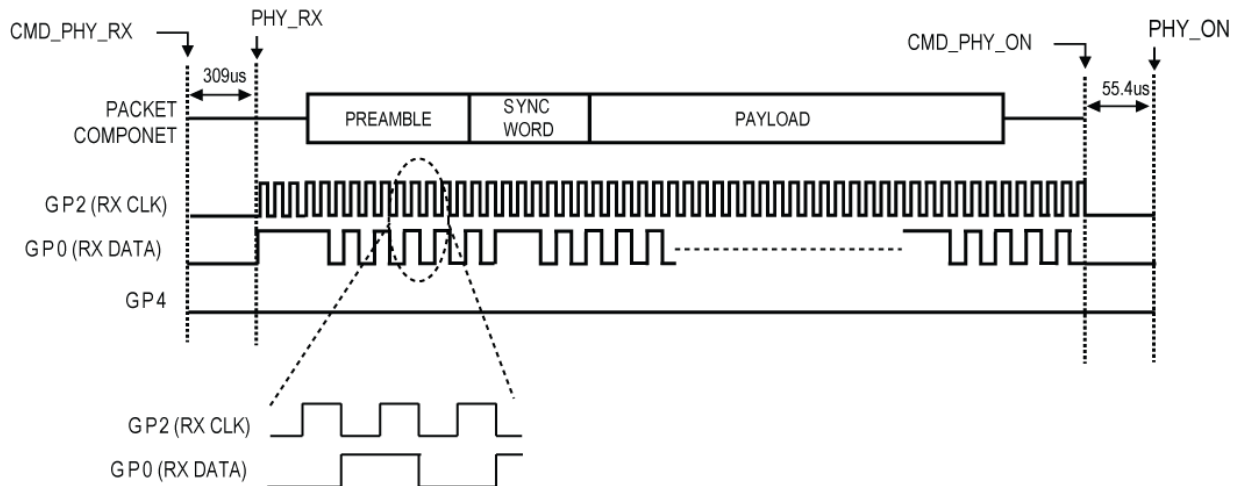
The data clock is enabled at the GP2 output together with the data signal at the DR\_GP0 output while in the PHY\_RX state. The IRQ\_GP3 is functional and can be used to interrupt on preamble or sync word.

### Mode 1: RX packet framing (GPIO\_CONTROL = 0xA1)

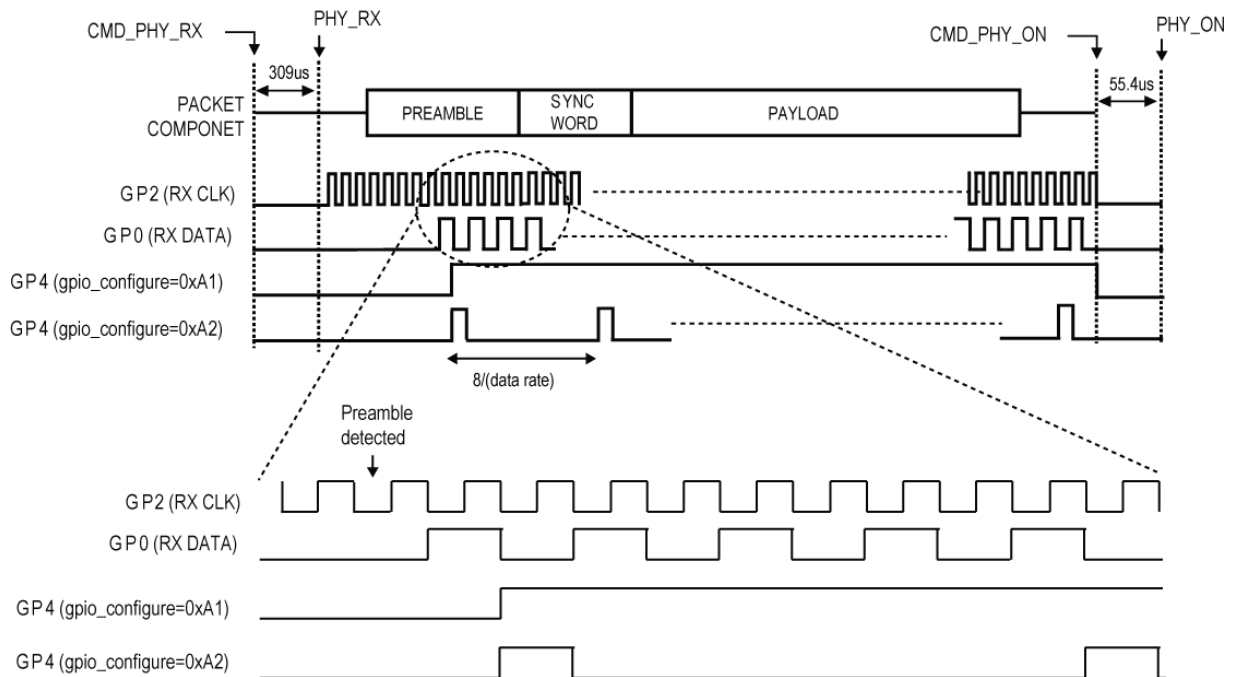
The data clock is enabled at the GP2 output together with the data signal at the DR\_GP0 output while in the PHY\_RX state. The IRQ\_GP3 is functional and can be used to interrupt on preamble or sync word. The GP4 output goes high when the sync word is detected and stays high for the number of bytes indicated by the PACKET\_LENGTH\_MAX setting. This can be used to frame the received payload data.

### Mode 2: RX byte framing (GPIO\_CONTROL = 0xA2)

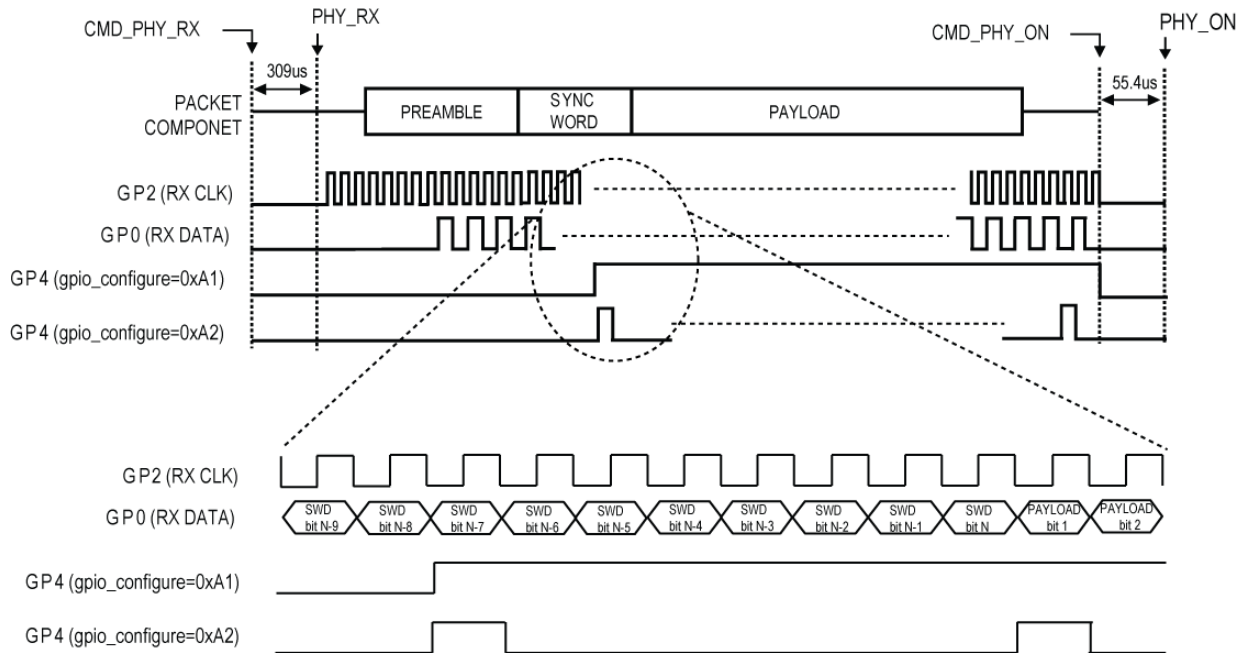
The data clock is enabled at the GP2 output together with the data signal at the DR\_GP0 output while in the PHY\_RX state. The IRQ\_GP3 is functional and can be used to interrupt on preamble or sync word. The GP4 output goes high when the sync word is detected for one bit period. It then stays low for seven bit periods before going high again for one bit period. This repeats until the end of the packet is reached (indicated by the PACKET\_LENGTH\_MAX setting).



**Figure 17: SPORT Mode Receive, data\_mode = 1 or 2**  
**GPIO\_CONFIGURE = 0xA0, 0xA3, 0xA6**



**Figure 18: SPORT Mode Receive, data\_mode = 1**  
**GPIO\_CONFIGURE = 0xA1, 0xA2, 0xA4, 0xA5, 0xA7, 0xA8**



**Figure 19: SPORT Mode Receive data\_mode = 2**  
**GPIO\_CONFIGURE = 0xA1, 0xA2, 0xA4, 0xA5, 0xA7, 0xA8**

## Transmit Bit Latencies in SPORT mode

The transmit bit latency is the time from the sampling of a bit by the transmit data clock on GP2 to when that bit appears at the RF output. There is no transmit bit latency when using 2FSK/MSK modulation. The latency when using GFSK/GMSK modulation is two bits. It is important that the host processor keeps the UHF transceiver in the PHY\_TX state for two bit periods after the last data bit is sampled by the data clock to account for this latency when using GMSK/GFSK modulation.

## Interrupts in Sport Mode

In sport mode, the interrupts from INTERRUPT\_SOURCE\_1 are all available. However, only INTERRUPT\_PREAMBLE\_DETECT and INTERRUPT\_SYNC\_DETECT are available from INTERRUPT\_SOURCE\_0. A second interrupt pin is provided on GP4, which gives a dedicated SPORT mode interrupts on either preamble or sync word detection.

## Interrupt Generation

The UHF transceiver uses a highly flexible, powerful interrupt system with support for MAC level interrupts and PHY level interrupts. Each interrupt source has a corresponding mask bit that needs to be set for the interrupt source to be enabled. When an enabled interrupt occurs, both the IRQ\_GP3 pin will go high and the interrupt bit of the status word will be set to logic 1. The host processor can either use the IRQ\_GP3 pin or the status word to check for an interrupt. After an interrupt is asserted, the UHF transceiver continues operations unaffected, unless it is directed to do otherwise by the host microprocessor. An outline of the interrupt source and mask system is shown in Figure 21.

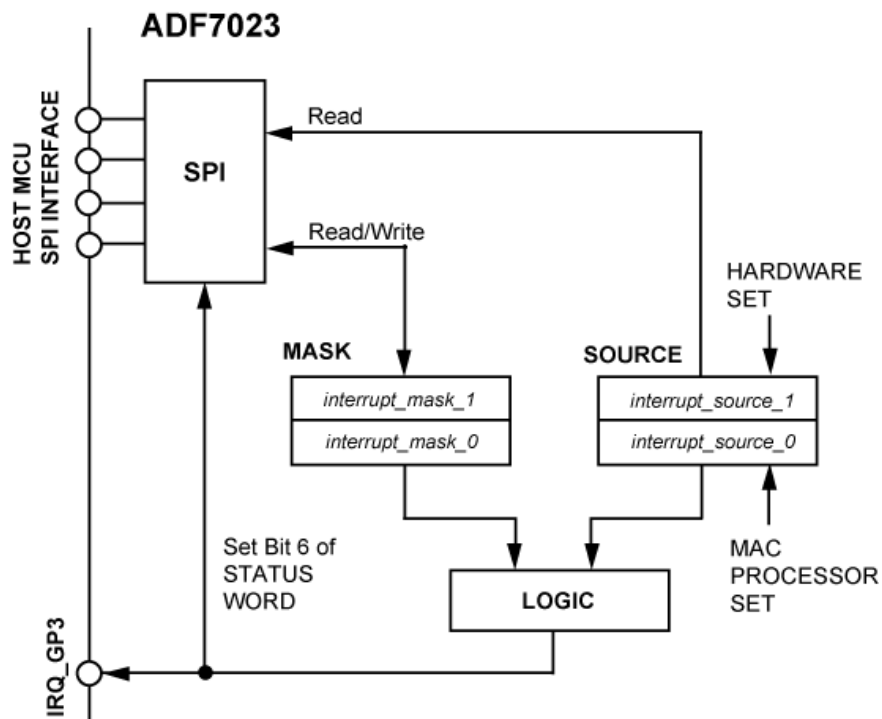


Figure 20: Overview of Interrupt Source and Mask Setup

## Interrupt Setup

MAC level interrupts can be enabled by writing logic 1 to the relevant bits in the INTERRUPT\_MASK\_0 register (Address 0x100).

PHY level interrupts can be enabled by writing logic 1 to the relevant bits of INTERRUPT\_MASK\_1 register (Address 0x101).

The structure of these two registers is given in Table 95 and Table 96.

**Table 95: Structure of the Interrupt Mask Register 0**  
 (Interrupt\_Mask\_0, Address 0x100)

Bit	Name	Function
7	INTERRUPT_NUM_WAKEUPS	Interrupt when the number of WUC wakeups (NUMBER_OF_WAKEUPS[15:0]) has reached the threshold (NUMBER_OF_WAKEUPS_IRQ_THRESHOLD[15:0]) 1: interrupt enabled 0: interrupt disabled
6	INTERRUPT_SWM_RSSI_DET	Interrupt when the measured RSSI during smart wake mode has exceeded the RSSI threshold value (SWM_RSSI_THRESH, Address 0x108) 1: interrupt enabled 0: interrupt disabled
5	INTERRUPT_AES_DONE	Interrupt when an AES encryption or decryption command is complete; available only when the AES firmware module has been loaded to the ADF7023 program RAM 1: interrupt enabled; 0: interrupt disabled
4	INTERRUPT_TX_EOF	Interrupt when a packet has finished transmitting 1: interrupt enabled; 0: interrupt disabled
3	INTERRUPT_ADDRESS_MATCH	Interrupt when a received packet has a valid address match 1: interrupt enabled; 0: interrupt disabled
2	INTERRUPT_CRC_CORRECT	Interrupt when a received packet has the correct CRC 1: interrupt enabled; 0: interrupt disabled
1	INTERRUPT_SYNC_DETECT	Interrupt when a qualified sync word has been detected in the received packet 1: interrupt enabled; 0: interrupt disabled
0	INTERRUPT_PREAMBLE_DETECT	Interrupt when a qualified preamble has been detected in the received packet 1: interrupt enabled; 0: interrupt disabled



**Table 96: Structure of the Interrupt Mask Register 1**  
(Interrupt\_Mask\_1, Address 0x101)

Bit	Name	Function
7	BATTERY_ALARM	Interrupt when the battery voltage has dropped below the threshold value (BATTERY_MONITOR_THRESHOLD_VOLTAGE, Address 0x32D) 1: interrupt enabled; 0: interrupt disabled
6	CMD_READY	Interrupt when the communications processor is ready to load a new command; mirrors the CMD_READY bit of the status word 1: interrupt enabled; 0: interrupt disabled
5	Reserved	Reserved
4	WUC_TIMEOUT	Interrupt when the WUC has timed out 1: interrupt enabled; 0: interrupt disabled
3	Reserved	Reserved
2	Reserved	Reserved
1	SPI_READY	Interrupt when the SPI is ready for access 1: interrupt enabled; 0: interrupt disabled
0	CMD_FINISHED	Interrupt when the communications processor has finished performing a command 1: interrupt enabled; 0: interrupt disabled

## Determining the Interrupt Source

In the case of an interrupt condition, the interrupt source can be determined by reading the INTERRUPT\_SOURCE\_0 register (Address 0x336) and the INTERRUPT\_SOURCE\_1 register (Address 0x337). The bit that corresponds to the relevant interrupt condition will be high. The structure of these two registers is shown in Table 97 and Table 98.

Following an interrupt condition, the host processor should clear the relevant interrupt flag, so that further interrupts will assert the IRQ\_GP3 signal. This is performed by writing logic 1 to the bit that is high in either the INTERRUPT\_SOURCE\_0 or INTERRUPT\_SOURCE\_1 register. If multiple bits in the interrupt source registers are high, they can be cleared individually or altogether by writing logic 1 to them. The IRQ\_GP3 signal will go low when all the interrupt source bits are cleared.

As an example, take the case where a battery\_alarm interrupt has occurred in the INTERRUPT\_SOURCE\_1 register. The host processor should:

- Read the interrupt source registers. In this example if none of the interrupt flags in INTERRUPT\_SOURCE\_0 were enabled, only INTERRUPT\_SOURCE\_1 needs to be read.
- Clear the interrupt by writing 0x80 (or 0xFF) to the INTERRUPT\_SOURCE\_1 register.
- Respond to the interrupt condition

**Table 97: Structure of Interrupt Source Register 0**  
(Interrupt Source 0, Address 0x336)

Bit	Name	Function
7	INTERRUPT_NUM_WAKEUPS	Asserted when the number of WUC wake-ups (NUMBER_OF_WAKEUPS[15:0]) has reached the threshold (NUMBER_OF_WAKEUPS_IRQ_THRESHOLD[15:0])
6	INTERRUPT_SWM_RSSI_DET	Asserted when the measured RSSI during smart wake mode has exceeded the RSSI threshold value (SWM_RSSI_THRESH, Address 0x108)
5	INTERRUPT_AES_DONE	Asserted when an AES encryption or decryption command is complete; available only when the AES firmware module has been loaded to the ADF7023 program RAM
4	INTERRUPT_TX_EOF	Asserted when a packet has finished transmitting (packet mode only)
3	INTERRUPT_ADDRESS_MATCH	Asserted when a received packet has a valid address match (packet mode only)
2	INTERRUPT_CRC_CORRECT	Asserted when a received packet has the correct CRC (packet mode only)
1	INTERRUPT_SYNC_BYTE_DETECT	Asserted when a qualified sync word has been detected in the received packet
0	INTERRUPT_PREAMBLE_DETECT	Asserted when a qualified preamble has been detected in the received packet

**Table 98: Structure of Interrupt Source Register 1**  
(Interrupt Source 1, Address 0x337)

Bit	Name	Function
7	BATTERY_ALARM	Asserted when the battery voltage has dropped below the threshold value (BATTERY_MONITOR_THRESHOLD_VOLTAGE, Address 0x32D)
6	RC_READY	Asserted when the communications processor is ready to load a new command; mirrors the CMD_READY bit of the status word
5	Reserved	Reserved
4	WUC_TIMEOUT	Asserted when the WUC has timed out
3	Reserved	Reserved
2	Reserved	Reserved
1	SPI_READY	Asserted when the SPI is ready for access
0	RC_ERROR	Asserted when the communications processor has finished performing a command

## UHF Transceiver Memory Map

This section describes the various memory locations used by the UHF transceiver. The radio control, packet management, and smart wake mode capabilities of the part are realized through the use of an integrated RISC processor, which executes instructions stored in the embedded program ROM. There is also a local RAM, subdivided into three sections, that is used as a data packet buffer, both for transmitted and received data (packet RAM), and for storing the radio and packet management configuration (BBRAM and MCR). The RAM addresses of these memory banks are 11 bits long.

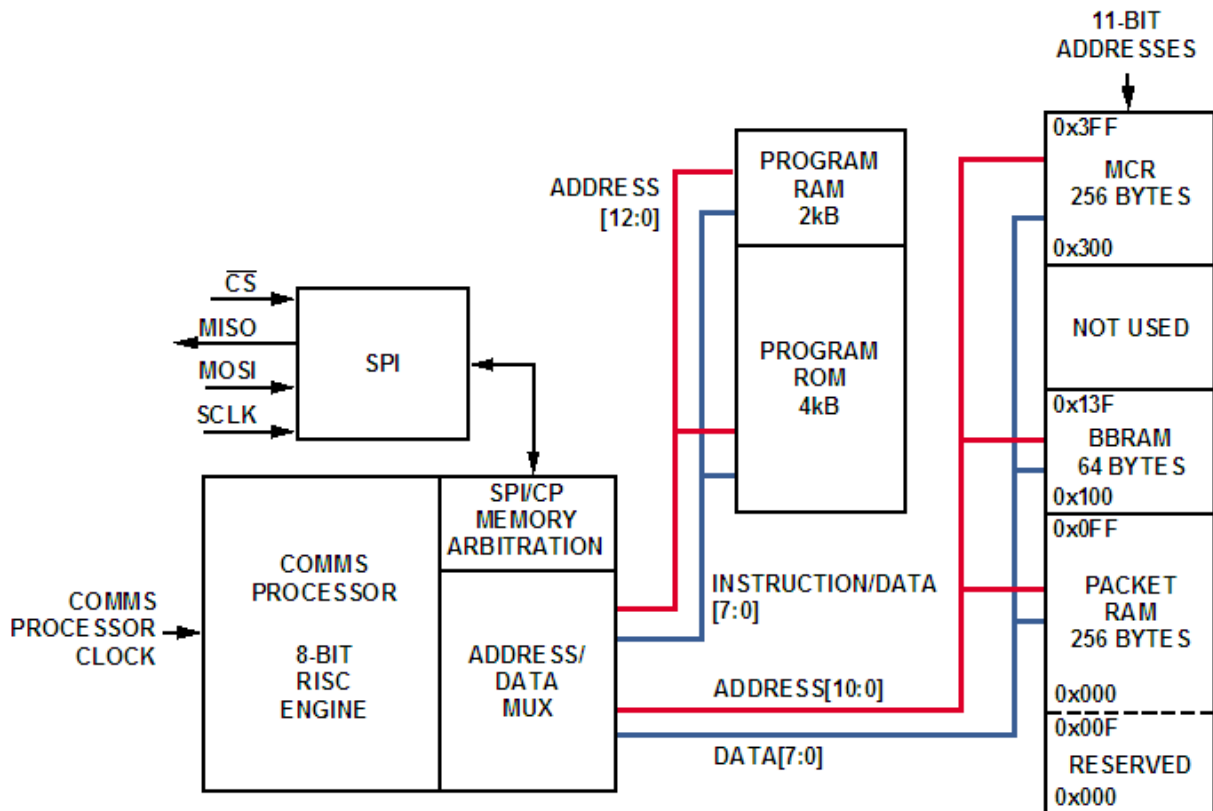


Figure 21: UHF TRANSCEIVER Memory Map

Note: More details of the BBRAM registers are available in Table 119.

Note: More details of the MCR registers are available Table 120.

### BBRAM (Battery Back Up RAM)

The battery backup RAM (BBRAM) consists of 64 bytes of memory space (0x100 to 0x13F). The BBRAM contains the main radio and packet management registers used to configure the radio. On application of battery power to the UHF transceiver for the first time, the entire BBRAM should be initialized by the host processor with the appropriate settings. After the BBRAM has been written to, the **CMD\_CONFIG\_DEV** command should be issued to update the radio and communications processor with the current BBRAM

settings. The CMD\_CONFIG\_DEV command can be issued in the PHY\_OFF state or the PHY\_ON state only.

The BBRAM is used to maintain settings needed at wake-up from sleep mode by the wake-up controller. Upon wake-up from sleep, in smart wake mode, the BBRAM contents are read by the on-chip processor to recover the packet management and radio parameters

## **MCR (Modem Configuration RAM)**

The MCR consists of 256 bytes of memory space (0x300 to 0x3FF). The MCR contains the various registers used for direct control or observation of the physical layer radio block of the UHF transceiver. Contents of MCR are lost during PHY\_SLEEP state.

## **Program ROM**

The program ROM consists of 4kbytes of non-volatile memory. It contains the firmware code for radio control, packet management and smart wake mode

## **Program RAM**

The program RAM consists of 2kbytes of volatile memory. This memory space is used for various software modules, such as AES encryption, IR calibration and Reed Solomon coding, which are available from Analog Devices. The software modules are downloaded to the program RAM memory space over the SPI by the host microprocessor.

## **Packet RAM**

The Packet RAM consists of 256 bytes of memory space divided into two sections: 240 bytes for storage of data from valid received packets and packet data to be transmitted and 16 bytes Reserved for use by the on-chip communications processor.

The communications processor will store received payload data at the memory location indicated by the value of the RX\_BASE\_ADR register (Address 0x125). The value of the TX\_BASE\_ADR register (Address 0x124) determines the start address of data to be transmitted by the communications processor.

This memory can be arbitrarily assigned to store single or multiple transmit or receiver packets both with and without overlap. The RX\_BASE\_ADR value should be chosen to ensure that there is enough allocated packet RAM space for the maximum receiver payload length.

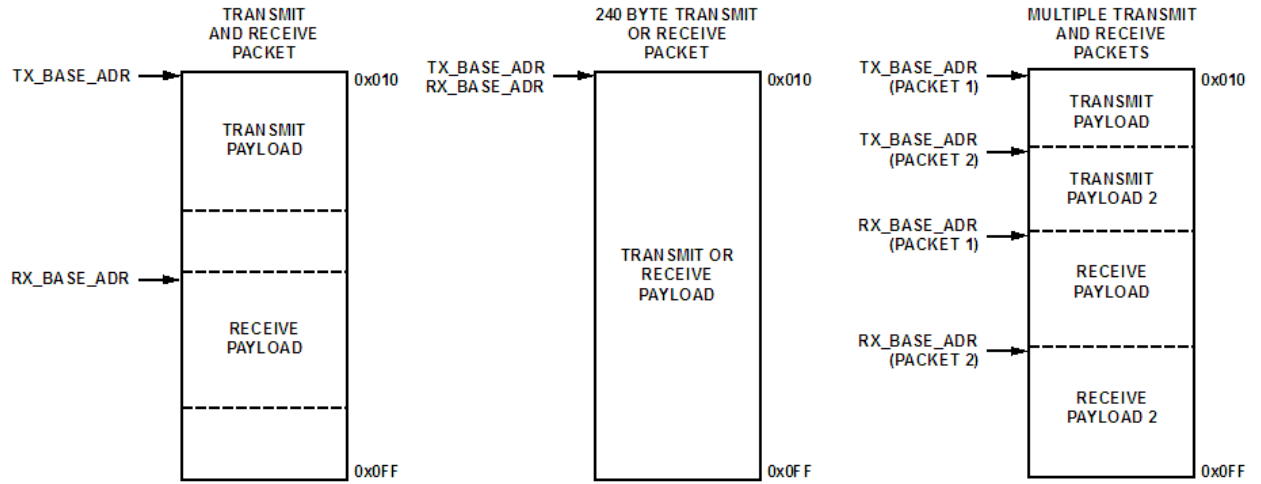


Figure 22: Example Packet RAM Configurations: Tx and Rx Packet Address Pointers

# UHF Transceiver SPI Interface

## General Characteristics

The UHF transceiver (ADF7023 die) is configured via a 4 wire SPI interface, using internal signals SCLK, MISO, MOSI and CS. The UHF transceiver always acts as a slave. Figure 23 describes the interdie connections between the UHF transceiver and the Cortex die. The diagram also shows the direction of the signal flow for each pin.

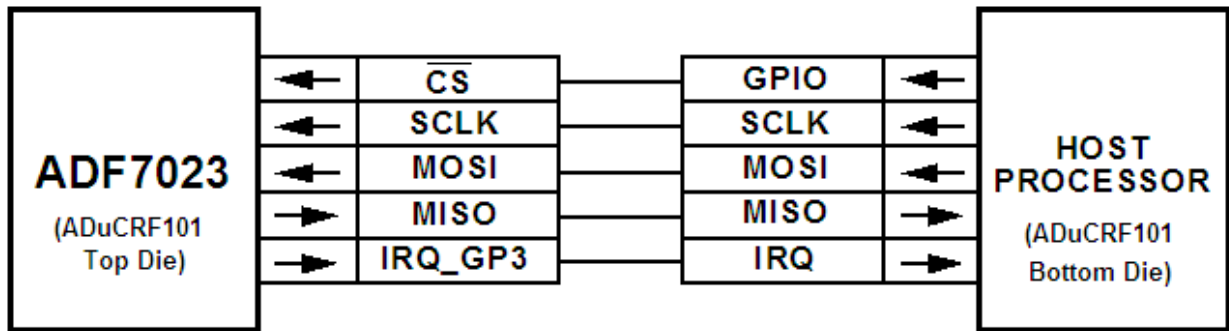


Figure 23: SPI Inter Die Connections

The SPI interface is active and the MISO output enabled only while the  $\overline{\text{CS}}$  input is low. The interface uses a word length of 8 bits, which is compatible with SPI of the bottom die. The data transfer through the SPI interface occurs with the most significant bit first. The MOSI input is sampled at the rising edge of SCLK. As commands or data are shifted in from the MOSI input at the SCLK rising edge, the status word or data is shifted out at the MISO pin synchronous with the SCLK clock falling edge. If CS is brought low, the MSB of the status word appears on the MISO output without the need for a rising clock edge on the SCLK input.

## Command Access

The UHF Transceiver is controlled through commands. Command words are single byte instructions, which control the state transitions of the radio and access to various memory locations. The communications processor handles commands with the CMD prefix, whereas memory access commands with the SPI prefix are handled independently. Thus SPI commands can be issued independent of the state of the radio.

A command is initiated by bringing  $\overline{\text{CS}}$  low and shifting in the command word over the SPI. All commands are executed after CS goes high again or at the next positive edge at the SCLK input (in the case of a memory access command). The  $\overline{\text{CS}}$  input must be brought high again once a command, with all its parameters, has been shifted into the UHF Transceiver in order to enable the recognition of successive command words. this is because a single command can be issued only during a  $\overline{\text{CS}}$  low period ( with the exception of a double NOP command)

The first byte returned on the MISO when writing a command should be ignored.

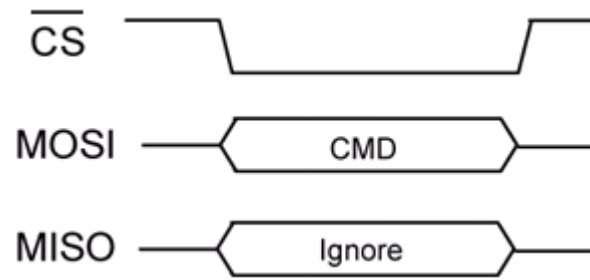


Figure 24: Command Write (no parameters)

## Status Word

The execution of certain commands by the communications processor may take several instruction cycles, during which the communications processor is busy. Prior to issuing a radio controller command it is therefore necessary to read the status word in order to determine if the UHF Transceiver is ready to accept a new command.

The status word of the UHF Transceiver is automatically returned over the MISO each time a byte is transferred over the MOSI. Shifting in double SPI\_NOP commands, will cause the status word to be shifted out. The meaning of the various bit fields is illustrated in Table 99.

The FW\_STATE variable can be used to read the current state of the communications processor and is described in Table 102. If it is busy performing an action or state transition, FW\_STATE is busy. The FW\_STATE variable also indicates the current state of the radio.

The SPI\_READY variable is used to indicate when the SPI is ready for access. The CMD\_READY variable is used to indicate when the communications processor is ready to accept a new command. The status word should be polled and the CMD\_READY bit examined before issuing a command to ensure that the communications processor is ready to accept a new command. It is not necessary to check the CMD\_READY bit before issuing a SPI memory access command. It is possible to queue one command while the communications processor is busy. The UHF transceiver interrupt handler can also be configured to generate an interrupt signal on IRQ\_GP3 when the communications processor is ready to accept a new command (CMD\_READY in the INTERRUPT\_SOURCE\_1 register (Address 0x337)) or when it has finished processing a command (CMD\_FINISHED in the INTERRUPT\_SOURCE\_1 register (Address 0x337)).

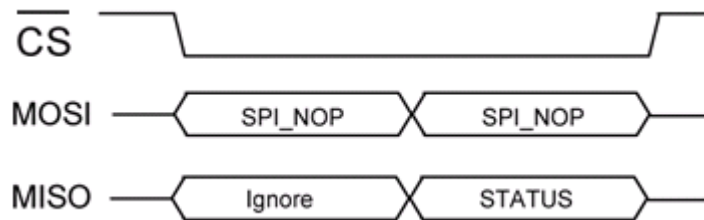


Figure 25: Reading the Status Word Using a Double SPI\_NOP Command



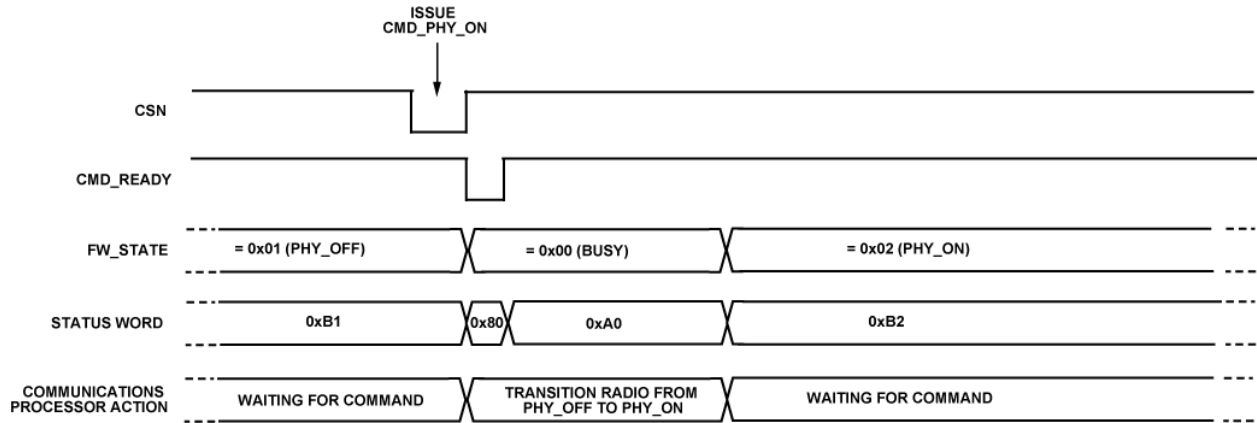
Table 99: SPI Status Word

Bit	Name	Description																										
7	SPI_READY	1: SPI is ready for access 0: SPI is not ready for access																										
6	IRQ_STATUS	1: Pending interrupt condition ( mirrors IRQ_GP3 pin) 0: No pending interrupt condition																										
5	CMD_READY	1: Radio Controller is ready to receive a command 0: Radio Controller is not ready to receive a command,																										
4 to 0	FW_STATE	Inticates the current state of the MAC processor																										
		<table><tr><th>Value</th><th>State</th></tr><tr><td>0x00</td><td>Busy. Performing a state transition</td></tr><tr><td>0x05</td><td>Performing CMD_GET_RSSI</td></tr><tr><td>0x06</td><td>PHY_SLEEP</td></tr><tr><td>0x07</td><td>Performing CMD_IR_CAL</td></tr><tr><td>0x08</td><td>Performing CMD_AES_DECRYPT_INIT</td></tr><tr><td>0x09</td><td>Performing CMD_AES_DECRYPT</td></tr><tr><td>0x0A</td><td>Performing CMD_AES_ENCRYPT</td></tr><tr><td>0x11</td><td>PHY_OFF</td></tr><tr><td>0x12</td><td>PHY_ON</td></tr><tr><td>0x13</td><td>PHY_RX</td></tr><tr><td>0x14</td><td>PHY_TX</td></tr><tr><td>0x0F</td><td>Initializing</td></tr></table>	Value	State	0x00	Busy. Performing a state transition	0x05	Performing CMD_GET_RSSI	0x06	PHY_SLEEP	0x07	Performing CMD_IR_CAL	0x08	Performing CMD_AES_DECRYPT_INIT	0x09	Performing CMD_AES_DECRYPT	0x0A	Performing CMD_AES_ENCRYPT	0x11	PHY_OFF	0x12	PHY_ON	0x13	PHY_RX	0x14	PHY_TX	0x0F	Initializing
		Value	State																									
		0x00	Busy. Performing a state transition																									
		0x05	Performing CMD_GET_RSSI																									
		0x06	PHY_SLEEP																									
		0x07	Performing CMD_IR_CAL																									
		0x08	Performing CMD_AES_DECRYPT_INIT																									
		0x09	Performing CMD_AES_DECRYPT																									
		0x0A	Performing CMD_AES_ENCRYPT																									
		0x11	PHY_OFF																									
		0x12	PHY_ON																									
		0x13	PHY_RX																									
		0x14	PHY_TX																									
0x0F	Initializing																											

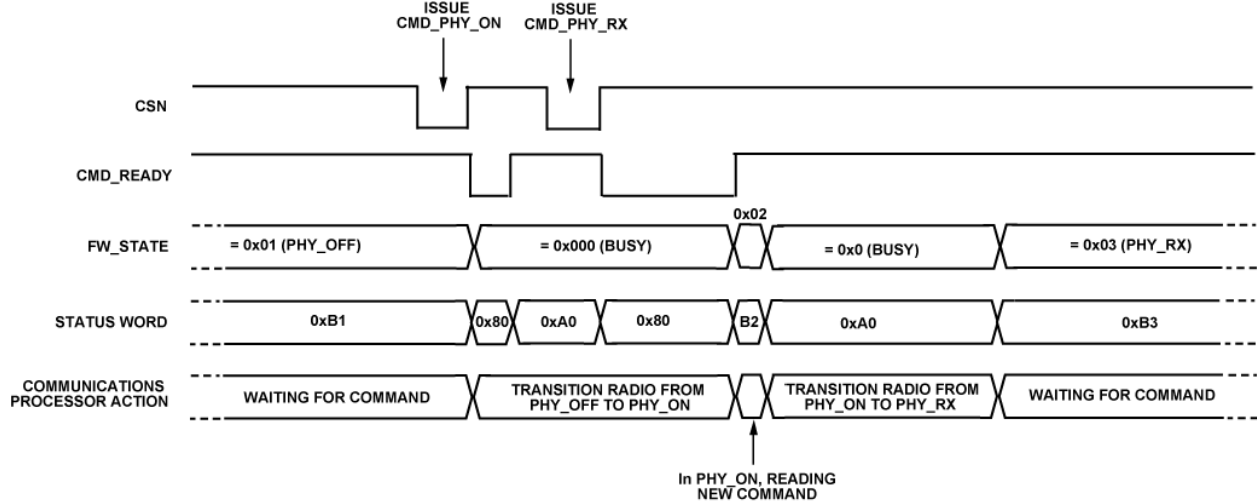
## Command Queuing

The CMD\_READY status bit is used to indicate that the command queue used by the communications processor is empty. The queue is one command deep. The FW\_STATE bit is used to indicate the state of the communications processor. The operation of the status word and these bits is illustrated in Figure 26 when a CMD\_PHY\_ON command is issued in PHY\_OFF.

The operation of the status word when a command is being queued is illustrated in Figure 27, when a CMD\_PHY\_ON command is issued in the PHY\_OFF state followed quickly by a CMD\_PHY\_RX command. The CMD\_PHY\_RX command is issued while the FW\_STATE is busy (i.e. transitioning between PHY\_OFF and PHY\_ON), but the CMD\_READY bit is high indicating that the command queue is empty. Once the CMD\_PHY\_RX command has been issued the CMD\_READY bit transitions to a logic low indicating the command queue is full. Once the PHY\_OFF to PHY\_ON transition is finished the PHY\_RX command is processed immediately by the communications processor and the CMD\_READY bit goes high, indicating that the command queue is empty and another command may be issued.



**Figure 26: Operation of the CMD\_READY and FW\_STATE Bits  
PHY\_OFF to PHY\_ON**



**Figure 27: Command Queuing and Operation of the CMD\_READY and FW\_STATE Bits  
PHY\_OFF to PHY\_ON and then to PHY\_RX**

## Memory Access

Memory locations are accessed by invoking the relevant SPI command. An 11 bit address is used to identify registers or locations in the memory space. The most significant 3 bits of the address are incorporated into the command by appending them as the LSBs of the command word. Figure 28 illustrates the command, address and data partitioning. The various SPI memory access commands are different depending on the memory location being accessed. This is described in Table 100.

An SPI command should only be issued only if the SPI\_READY bit of the status word bit is high. The UHF transceiver interrupt handler can also be configured to generate an interrupt signal on IRQ\_GP3 when the SPI\_READY bit is high (SPI\_READY, Address 0x337: interrupt\_source\_1).

Also, an SPI command should not be issued while the communications processor is initializing (FW\_STATE= 0x0F). SPI commands can be issued in any other

communications processor state including the busy state (FW\_STATE= 0x00). This allows the UHF transceiver memory to be accessed while the radio is transitioning between states.

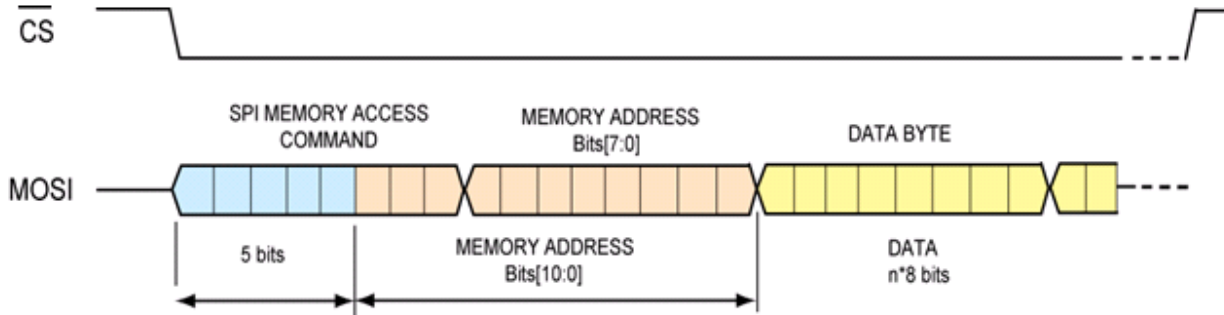


Figure 28: SPI Memory Access Command/Address Format

Table 100: Summary of SPI Memory Access Commands

SPI Command	Command Value	Description
<b>SPI_MEM_WR</b>	= 0x18 (packet RAM) = 0x19 (BBRAM) = 0x1B (MCR) = 0x1E (program RAM)	Write data to BBRAM, MCR or packet RAM memory sequentially. An 11 bit address is used to identify memory locations. The most significant 3 bits of the address are incorporated in to the command (xxxb). This command is followed by the remaining 8 bits of the address, which are subsequently followed by the data bytes to be written.
<b>SPI_MEM_RD</b>	= 0x38 (packet RAM) = 0x39 (BBRAM) = 0x3B (MCR)	Read data from BBRAM, MCR or packet RAM memory sequentially. An 11 bit address is used to identify memory locations. The most significant 3 bits of the address are incorporated in to the command (xxxb). This command is followed by the remaining 8 bits of the address, which is subsequently followed by the appropriate number of SPI_NOP commands.
<b>SPI_MEMR_WR</b>	= 0x08 (packet RAM) = 0x09 (BBRAM) = 0x0B (MCR)	Write data to BBRAM/MCR or Packet RAM memory nonsequentially.
<b>SPI_MEMR_RD</b>	= 0x28 (packet RAM) = 0x29 (BBRAM) = 0x2B (MCR)	Read data from BBRAM/MCR or Packet RAM at random.
<b>SPI_NOP</b>	= 0xFF	No operation. Use for dummy writes when polling the status_word. Also used as dummy data when performing a memory read.

## Block Write

MCR, BBRAM and Packet RAM memory locations can be written to in block format using the SPI\_MEM\_WR command. The SPI\_MEM\_WR command code is 00011xxxb, where xxxb represent Bits [10:8] of the first 11-bit address. If more than one data byte is written, the write address is automatically incremented for every byte sent until  $\overline{CS} = 1$  terminates the command. The maximum block write for the MCR, packet RAM and BBRAM memories are 256 bytes, 256 bytes and 64 bytes respectively. These maximum block-write lengths should not be exceeded. Refer to Figure 29 for more details.

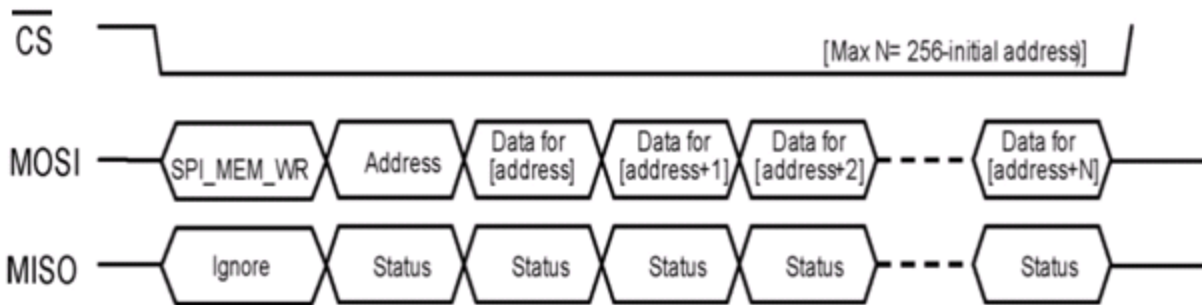


Figure 29: Memory (MCR, BBRAM or Packet RAM) Block Write

### Block Write Example

Write 0x00 to the ADC\_CONFIG\_HIGH register (memory locations 0x35A).

1. CS line should be set low to initiate SPI access
2. The first five bits of the SPI\_MEM\_WR command are 00011.
3. The 11-bit address of ADC\_CONFIG\_HIGH is 01101011010 (0x35Ah).
4. The first byte sent is 00011011 or 0x1B. (SPI\_MEM\_WR command and the first most significant bits of the address)
5. The second byte sent is 01011010 or 0x5A. (remaining 8-bits of the register address)
6. The third byte sent is 0x00. (data byte follows)
7. CS=1 terminates the command

Thus, 0x1B5A00 is written to the part.

### Random Address Write

MCR, BBRAM and Packet RAM memory locations can be written to in random address format using the SPI\_MEMR\_WR command. The SPI\_MEMR\_WR command code is 00001xxxb, where xxxb represent bits [10:8] of the 11-bit address. The lower 8 bits of the address should follow this command and then the data byte to be written to the address. The lower 8 bits of the next address are entered followed by the data for that address until all required addresses within that block are written, as shown in Figure 30.

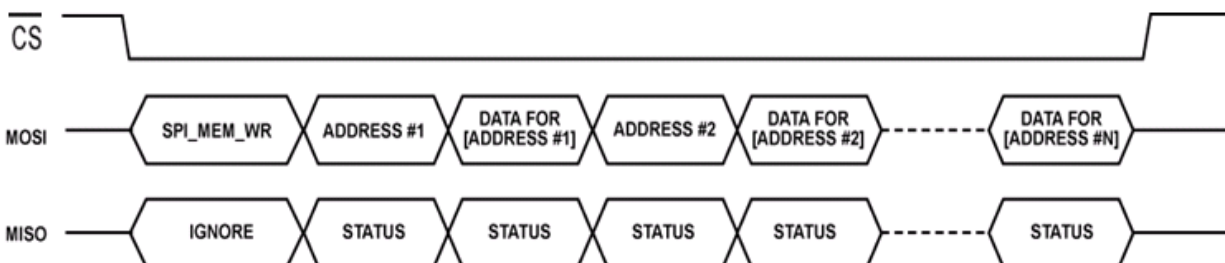


Figure 30: Memory (MCR, BBRAM or Packet RAM) Random Address Write

## Block Read

MCR, BBRAM and Packet RAM memory locations can be read from in block format using the SPI\_MEM\_RD command. The SPI\_MEM\_RD command code is 00111xxxb, where xxxb represent Bits [10:8] of the first 11-bit address. This command is followed by the remaining 8 bits of the address to be read and then two SPI\_NOP commands (dummy byte). The first byte available after writing the address should be ignored, with the second byte constituting valid data. If more than one data byte is to be read, the write address is automatically incremented for subsequent SPI\_NOP commands sent. Refer to Figure 31 for more details.

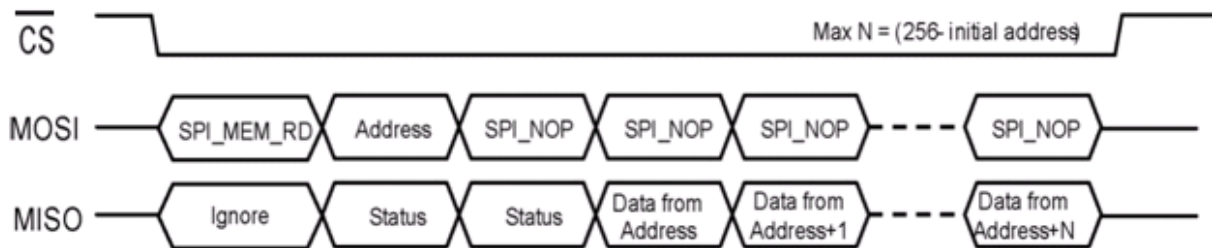


Figure 31: Memory (MCR, BBRAM or Packet RAM) Block Read

## Random Address Read

MCR, BBRAM and Packet RAM memory locations can be read from in a non-sequential manner using the SPI\_MEMR\_RD command. The SPI\_MEMR\_RD command code is 00101xxxb, where xxxb represent Bits [10:8] of the 11-bit address. This command is followed by the remaining 8 bits of the address to be written. Each subsequent address byte is then written. The last address byte to be written should be followed by two SPI\_NOP commands as shown in Figure 32. The data bytes from memory, starting at the first address location are available after the second status byte.

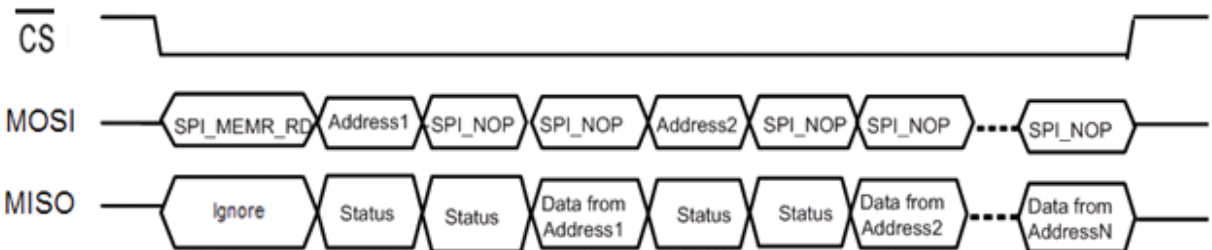


Figure 32: Memory (MCR, BBRAM or Packet RAM) Random Address Read

**Random Address Read Example**

Read the value stored in the ADC\_CONFIG\_HIGH register.

1. The first five bits of the SPI\_MEM\_RD command are 00111.
2. The 11-bit address of ADC\_CONFIG\_HIGH is 01101011010.
3. The first byte sent is 00111011 or 0x3B. (SPI\_MEM\_RD command and the first most significant bits of the address)
4. The second byte sent is 01011010 or 0x5A. . (remaining 8-bits of the register address)
5. The third byte sent is 0xFF (SPI\_NOP).
6. The fourth byte sent is 0xFF.

Thus, 0x3B5AFFFF is written to the part.

The value shifted out on the MISO line while the 4<sup>th</sup> byte is sent is the value stored in the ADC\_CONFIG\_HIGH register.

## SPI Memory Access Commands

A summary of the SPI memory access commands are given in Table 101 below.

**Table 101: Overview of SPI Memory Access Commands**

Command		Description
SPI_MEM_WR	00011xxxb	Write data to BBRAM/MCR or Packet RAM sequentially. An 11 bit address is used to identify memory locations. The most significant 3 bits of the address are incorporated in to the command (xxxb). This command is followed by the remaining 8 bits of the address.
SPI_MEM_RD	00111xxxb	Read data from BBRAM/MCR or Packet RAM sequentially. An 11 bit address is used to identify memory locations. The most significant 3 bits of the address are incorporated in to the command (xxxb). This command is followed by the remaining 8 bits of the address, which is subsequently followed by the appropriate number of SPI_NOP commands.
SPI_MEMR_WR	00001xxxb	Write data to BBRAM/MCR or Packet RAM at random.
SPI_MEMR_RD	00101xxxb	Read data from BBRAM/MCR or Packet RAM at random.
SPI_NOP	0xFFFFh	No operation. Use for dummy writes when polling the status_word. Used also as dummy data on the MOSI line when performing a memory read.

## Low Power Modes

The UHF transceiver has several low power modes to meet a wide variety of applications. These low power modes are implemented using a hardware wake-up controller (WUC), a firmware timer and the smart wake mode functionality of the on-chip communications processor. The hardware WUC is a low power wake up controller (WUC) which comprises of a 16-bit wake-up timer with a programmable pre-scaler. The 32.768 kHz RCOSC or XOSC provides the clock source for the timer.

The firmware timer is a software timer residing on the UHF transceiver. The firmware timer is used to count the number of WUC timeouts and so can be used to count the number of UHF transceiver wakeups. The WUC and the firmware timer therefore provide a real-time clock capability.

Utilizing the low power WUC and the firmware timer, the SWM firmware allows the UHF transceiver to wake up autonomously from sleep without intervention from the host microprocessor. During this wake-up period the UHF transceiver is controlled by the communications processor. This functionality allows carrier sense, packet sniffing and packet reception while the host processor is in sleep, thereby dramatically reducing overall system current consumption. The smart wake mode can then wake the host processor on an interrupt condition. An overview of the low power mode configuration is shown in Figure 33 and the register settings that are used for the various low power modes are described in Table 102.



Table 102: Settings for Low Power Mode

LPM	Memory Location	Register Name	Field	Description
Deep Sleep Modes	0x30D <sup>1</sup>	wuc_config_low	wuc_bbram_en	Set to 0: BBRAM contents are not retained during PHY_SLEEP Set to 1: BBRAM contents are retained during PHY_SLEEP
WUC	0x30C <sup>1</sup>	wuc_config_high	wuc_prescaler[2:0]	Sets the pre-scaler value of the WUC
WUC	0x30D <sup>1</sup>	wuc_config_low	wuc_rcosc_en	Enables the 32.768kHz RC OSC
WUC	0x30D <sup>1</sup>	wuc_config_low	wuc_xosc32k_en	Enables the 32.768kHz external OSC
WUC	0x30D <sup>1</sup>	wuc_config_low	wuc_xosc32k_clkssel	Sets the WUC clock source: 1: RC OSC selected 2: XOSC selected
WUC	0x30D <sup>1</sup>	wuc_config_low	wuc_arm	Enable to ensure device wakes from PHY_SLEEP on a WUC timeout
WUC	0x30E <sup>2</sup> 0x30F	wuc_value_high wuc_value_low	wuc_timer_value[15:0]	The WUC timer value. WUC interval (s) $= \frac{2^{(wuc\_prescaler+1)}}{32768}$
WUC	0x101	interrupt_mask_1	wuc_timeout	Enables the interrupt on WUC timeout
Firmware timer		interrupt_mask_0	interrupt_num_wakeups	Enabling this interrupt enables the firmware timer. Interrupt is set when the number_of wakeups count exceeds the threshold.
Firmware timer	0x102, 0x103	number_of_wakeups_0 number_of_wakeups_1	number_of_wakeups[15:0]	Number of ADF7023 wakeups.
Firmware timer	0x104, 0x105	number_of_wakeups_ irq_threshold_0 number_of_wakeups_ irq_threshold_1	number_of_wakeups_ irq_threshold[15:0]	Threshold for the number of ADF7023 wakeups. When exceeded the ADF7023 will exit the low power mode.
SWM	0x11A	mode_control	swm_en	Enables Smart Wake Mode
SWM	0x11A	mode_control	swm_rssi_qual_en	Enables RSSI pre-qualification in smart wake mode
SWM	0x108	swm_rssi_thresh	swm_rssi_thresh[7:0]	RSSI threshold for RSSI pre-qualification. RSSI threshold (dBm) = swm_rssi_thresh - 107
SWM	0x107	parmtime_divider	parmtime_divider[7:0]	Tick rate for the rx dwell timer.
SWM	0x106	rx_dwell_time	rx_dwell_time[7:0]	Time that ADF7023 remains awake during SWM. Receive dwell time (s) $= rx\_dwell\_time \times \frac{6.5MHz}{128 \times parmtime\_divider}$
SWM	0x100	interrupt_mask_0	interrupt_swm_rssi_det interrupt_preamble_detect interrupt_sync_detect interrupt_preamble_correct interrupt_address_match	Various interrupts that can be used in SWM

<sup>1</sup> It is necessary to write to both 0x30C and 0x30D registers in the following order:  
WUC\_CONFIG\_HIGH (0x30C) directly followed by writing to WUC\_CONFIG\_LOW (0x30D).

<sup>2</sup> It is necessary to write to both 0x30E and 0x30F registers in the following order:  
WUC\_VALUE\_HIGH (0x30E) directly followed by writing to WUC\_VALUE\_LOW (0x30F).

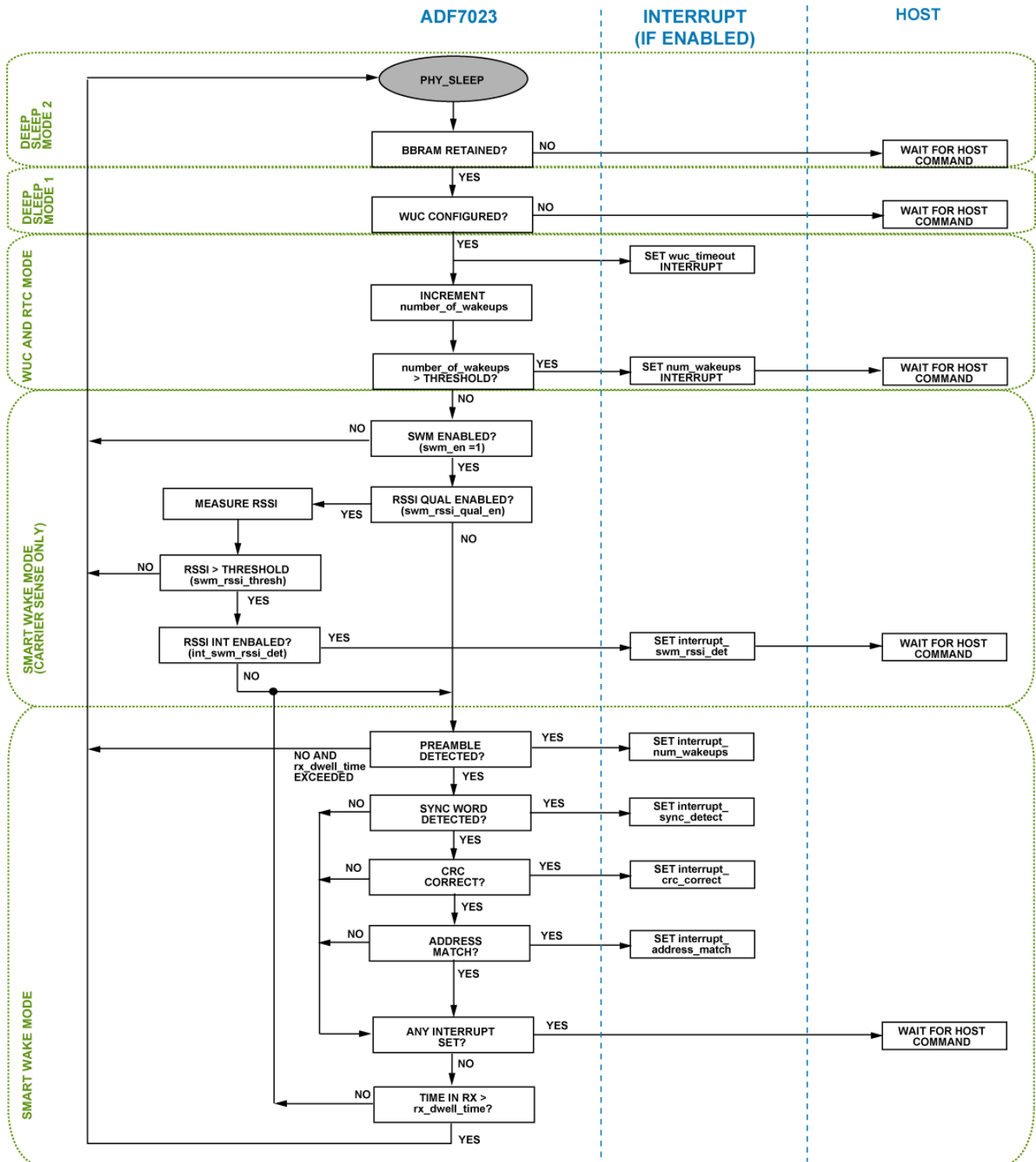


Figure 33: Low Power Mode Operation

## Example Low Power Modes

### Deep Sleep Mode 2

This mode is suitable for applications where the host processor is controlling the low power mode timing and where the lowest possible UHF transceiver sleep current is required. In this low power mode the UHF transceiver is in PHY\_SLEEP. The BBRAM contents are not retained. This low power mode is entered by issuing the CMD\_HW\_RESET command from any radio state. To wake the part from the PHY\_SLEEP state the CS line should be set low. The initialization routine after a CMD\_HW\_RESET should be followed as detailed in the Radio Control section.

### Deep Sleep Mode 1

This mode is suitable for applications where the host processor is controlling the low power mode timing and where the UHF transceiver configuration is retained during PHY\_SLEEP.

In this low power mode the UHF transceiver is in PHY\_SLEEP with the BBRAM contents retained. Before entering the PHY\_SLEEP state, WUC\_CONFIG\_LOW [3] (Address 0x30D) should be set to one to ensure that the BBRAM is retained. This low power mode is entered by issuing the CMD\_PHY\_SLEEP command from either the PHY\_OFF or PHY\_ON states. To exit the PHY\_SLEEP state the CS line can be set low. The CS low initialization routine should then be followed as detailed in the Radio Control section.

### WUC Mode

In this low power mode the hardware WUC is used to wake the UHF transceiver from the PHY\_SLEEP state after a user-defined duration. At the end of this duration, the UHF Transceiver can provide an interrupt to the host microprocessor. While the UHF Transceiver is in the PHY\_SLEEP state the host processor can optionally be in a deep sleep state to save power.

Before issuing the CMD\_PHY\_SLEEP command, the host processor should configure the WUC and set the firmware timer threshold to zero (number\_of\_wakeups\_irq\_threshold = 0). The WUC\_CONFIG\_LOW [3] (Address 0x30D) should be set to one to ensure that the BBRAM is retained. On issuing CMD\_PHY\_SLEEP the device then goes to sleep for a period until the hardware timer times out. At this point, the device wakes up, and if the WUC\_INTERRUPT or the INTERRUPT\_NUM\_WAKEUPS is enabled the device asserts the IRQ\_GP3 pin.

The operation of this low power mode is illustrated in Figure 34.

## WUC Mode with Firmware Timer

In this low power mode the WUC is used to periodically wake the UHF Transceiver from the PHY\_SLEEP state and the firmware timer is used count the number of WUC timeouts. The combination of the WUC and the firmware timer provide a real-time clock (RTC) capability.

The host processor should set up the WUC and the firmware timer before entering PHY\_SLEEP. The WUC\_BBRAM\_EN (Address 0x30D) should be set to 1 to ensure that the BBRAM is retained. The WUC can be configured to timeout at some standard time interval (for example 1 second, 60 seconds ). On issuing the CMD\_PHY\_SLEEP command, the device enters the PHY\_SLEEP state for a period until the hardware timer times out. At this point, the device wakes up, increments the 16-bit firmware timer (NUMBER\_OF\_WAKEUPS, Address 0x102 and Address 0x103) and if the WUC\_TIMEOUT is enabled (Address 0x101), the device asserts the IRQ\_GP3 pin. If the 16-bit firmware count is less than or equal to the user set value (NUMBER\_OF\_WAKEUPS\_IRQ\_THRESHOLD, Address 0x105), the device returns to the PHY\_SLEEP state. With this method, the firmware count (NUMBER\_OF\_WAKEUPS) equates to a real time interval.

When the firmware count exceeds a threshold (NUMBER\_OF\_WAKEUPS\_IRQ\_THRESHOLD) the UHF Transceiver asserts the IRQ\_GP3 pin, if the INTERRUPT\_NUM\_WAKEUPS bit is set (Address 0x100) and then enters the PHY\_OFF state. The operation of this low power mode is illustrated in Figure 35.

## Smart Wake Mode (Carrier Sense only)

In this low power mode the WUC, firmware timer and smart wake mode are used to implement periodic RSSI measurements on a particular channel ( that is, carrier sense). To enable this mode the WUC and firmware timer should be configured before entering the PHY\_SLEEP state. The WUC\_BBRAM\_EN (Address 0x30D) should be set to 1 to ensure that the BBRAM is retained. The RSSI measurement is enabled by setting SWM\_RSSI\_QUAL = 1 (MODE\_CONTROL, Address 0x11A) and setting SWM\_EN = 1 (MODE\_CONTROL, Address 0x11A). The INTERRUPT\_SWM\_RSSI\_DET (INTERRUPT\_MASK\_0, Address 0x100) should also be enabled. If the measured RSSI value is below the user defined threshold, set in the SWM\_RSSI\_THRESH register (Address 0x108), then the device will return to the PHY\_SLEEP state. If the RSSI measurement is greater than the SWM\_RSSI\_THRESH value, the device sets the INTERRUPT\_SWM\_RSSI\_DET interrupt to alert the host processor and waits in the PHY\_ON state for a host command. The operation of this low power mode is illustrated in Figure 36.

## Smart Wake Mode

In this low power mode the WUC, firmware timer and smart wake mode are employed to periodically listen for packets. To enable this mode the WUC and firmware timer should be configured and smart wake mode (SWM) enabled (SWM\_EN Address 0x11A), before entering the PHY\_SLEEP state. The WUC\_BBRAM\_EN (Address 0x30D) should be set to 1 to ensure that the BBRAM is retained. The RSSI pre-qualification can be optionally enabled (SWM\_RSSI\_QUAL = 1). When RSSI pre-qualification is enabled the UHF

Transceiver will only begin searching for preamble if the RSSI measurement is greater than the user-defined threshold.

The UHF Transceiver will be in the PHY\_RX state for a duration determined by the RX\_DWELL\_TIME setting (Address 0x106). If the UHF Transceiver detects preamble during the receive dwell time it will then search for sync word. If the sync word routine is detected the UHF Transceiver will load the received data to packet RAM and check for a CRC and address match, if enabled. If any of the receive packet interrupts have been set then the UHF Transceiver will return to the PHY\_ON state and wait for a host command.

If the UHF Transceiver has received preamble detection during the receive dwell time but the remainder of the received packet extends beyond the dwell time then the UHF Transceiver will extend the dwell time until all of the packet has been received or the packet has been recognized as invalid (for example there is an incorrect sync word).

This low power mode terminates when a valid packet interrupt has been received. Alternatively, this lower power can be terminated by using the firmware timer timeout. This can be useful if certain radio tasks (e.g. IR calibration) or microprocessor tasks need to be run periodically while in the low power mode.

The operation of this low power mode is illustrated in Figure 37.

### Exiting Low Power Mode

As described in Figure 33 the UHF transceiver wait for a host command on any of the termination conditions of the low power mode.

It is also possible to perform an asynchronous exit from low power mode using the following procedure:

1. Bring  $\overline{\text{CS}}$  low and wait until the MISO output goes high
2. Issue a CMD\_HW\_RESET command.

The host processor should then follow the initialization procedure after a CMD\_HW\_RESET command.

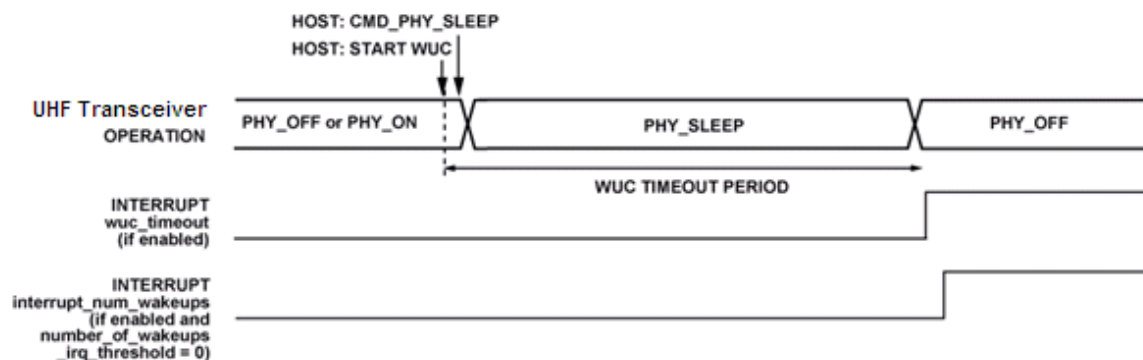


Figure 34: Low Power Mode Timing when using the WUC

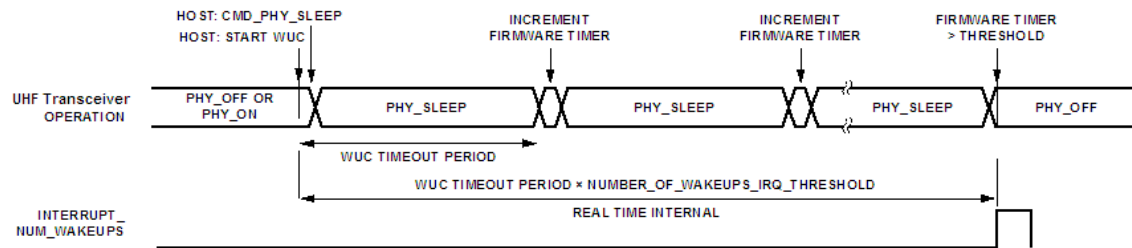


Figure 35: Low Power Mode Timing when using the WUC and the Firmware Timer

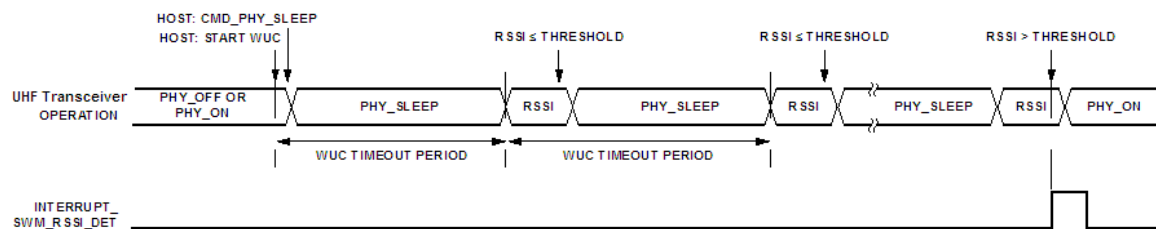


Figure 36: Low Power Mode Timing when using the WUC, the Firmware Timer and SWM with Carrier Sense Only

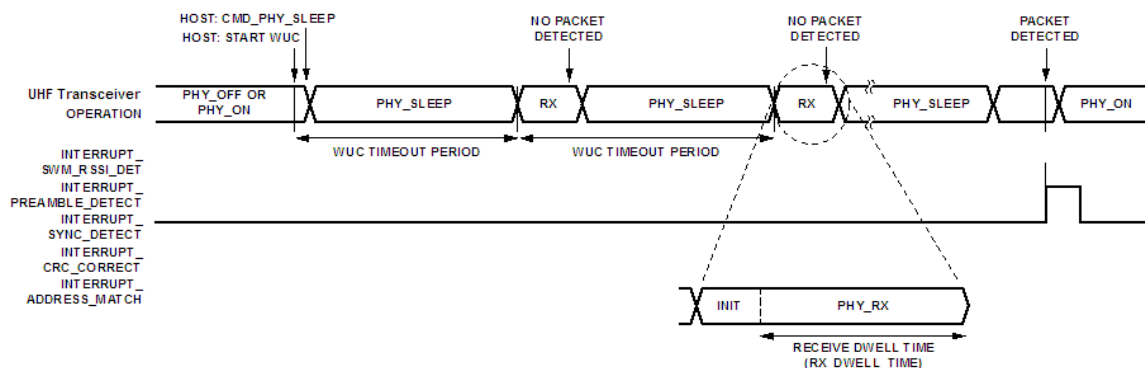
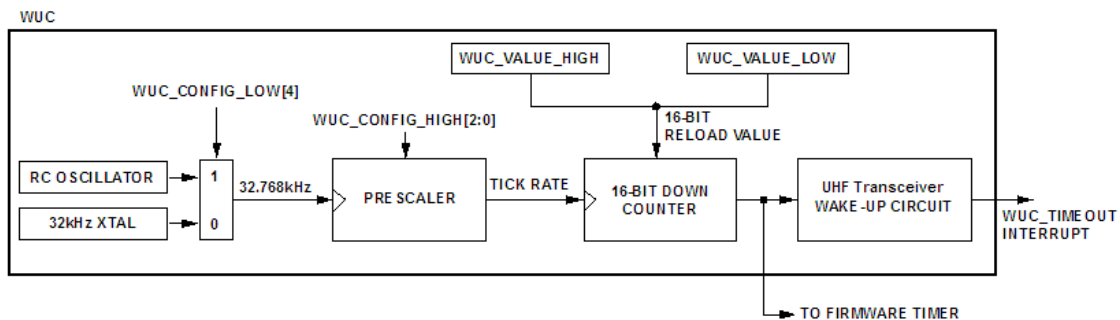


Figure 37: Low Power Mode Timing when using the WUC, the Firmware Timer and SWM

## Wake Up Controller (WUC) Setup

### Circuit Description

The UHF Transceiver features a low power wake-up controller comprising of a 16-bit wake-up timer with a 3-bit programmable pre-scaler as illustrated in Figure 38. The pre-scaler clock source can be configured to use either the 32.768 kHz internal RC oscillator (RCOSC) or the 32.768 kHz external oscillator (XOSC). This combination of programmable pre-scaler and 16-bit down counter gives a total hardware timer range of 30.52 $\mu$ s to 36.4 hours.



**Figure 38: Hardware Wake Up Controller (WUC)**

### Configuration and Operation

The hardware WUC is configured via the following registers:

1. WUC\_CONFIG\_HIGH (Address 0x30C)
2. WUC\_CONFIG\_LOW (Address 0x30D)
3. WUC\_VALUE\_HIGH (Address 0x30E)
4. WUC\_VALUE\_LOW (Address 0x30F)

The relevant fields of each register are detailed in Table 103. All four of these registers are write only.

The WUC should be configured as follows:

1. Clear all interrupts
2. Set required interrupts
3. Write to WUC\_CONFIG\_HIGH and WUC\_CONFIG\_LOW. Ensure that WUC\_ARM =1. Ensure WUC\_BBRAM\_EN =1 (retain BBRAM during PHY\_SLEEP). It is necessary to write to both registers together in the following order: WUC\_CONFIG\_HIGH directly followed by writing to WUC\_CONFIG\_LOW.
4. Write to WUC\_VALUE\_HIGH and WUC\_VALUE\_LOW. This configures the WUC\_TIMER\_VALUE [15:0] and thus the WUC timeout period. The timer will begin counting from the configured value once these registers have been written to. It is necessary to write to both registers together in the following order: WUC\_TIMER\_VALUE\_HIGH directly followed by writing to WUC\_VALUE\_LOW.

Table 103: WUC Register Settings

WUC Setting	Name	Description																											
WUC_VALUE_HIGH [7:0]	WUC_TIMER_VALUE [15:8]	WUC timervalue WUC Interval(s) = $WUC\_TIMER\_VALUE \times \frac{2^{(WUC\_PRESCALER + 1)}}{32768}$																											
WUC_VALUE_LOW [7:0]	WUC_TIMER_VALUE [7:0]	WUC timer value																											
WUC_CONFIG_HIGH [7:3]	Reserved	Set to 0																											
WUC_CONFIG_HIGH [2:0]	WUC_PRESCALER	<table> <tr> <th>wuc_prescaler</th><th>32.768kHz Divider</th><th>Tick Period</th></tr> <tr><td>000</td><td>1</td><td>30.52us</td></tr> <tr><td>001</td><td>4</td><td>122.1us</td></tr> <tr><td>010</td><td>8</td><td>244.1us</td></tr> <tr><td>011</td><td>16</td><td>488.3us</td></tr> <tr><td>100</td><td>128</td><td>3.91ms</td></tr> <tr><td>101</td><td>1034</td><td>31.25ms</td></tr> <tr><td>110</td><td>8192</td><td>250ms</td></tr> <tr><td>111</td><td>65536</td><td>2000ms</td></tr> </table>	wuc_prescaler	32.768kHz Divider	Tick Period	000	1	30.52us	001	4	122.1us	010	8	244.1us	011	16	488.3us	100	128	3.91ms	101	1034	31.25ms	110	8192	250ms	111	65536	2000ms
wuc_prescaler	32.768kHz Divider	Tick Period																											
000	1	30.52us																											
001	4	122.1us																											
010	8	244.1us																											
011	16	488.3us																											
100	128	3.91ms																											
101	1034	31.25ms																											
110	8192	250ms																											
111	65536	2000ms																											
WUC_CONFIG_LOW[7]	Reserved	Set to 0																											
WUC_CONFIG_LOW [6]	WUC_RCOSC_EN	1: Enable 0: Disable RCOSC32K																											
WUC_CONFIG_LOW [5]	WUC_XOSC32K_EN	1: Enable XOSC32K 0: Disable XOSC32K																											
WUC_CONFIG_LOW [4]	WUC_CLKSEL	Select the WUC timer clock source 1: RC 32.768 kHz oscillator; 0: external crystal oscillator																											
WUC_CONFIG_LOW [3]	WUC_BBAM_EN	1: Enable power to BBRAM during the PHY_SLEEPstate 0: Disable power to BBRAM during the PHY_SLEEP state																											
WUC_CONFIG_LOW [2:1]	Reserved	Set to 0																											
WUC_CONFIG_LOW [0]	WUC_ARM	1: Enable wake-up on WUC timeout event 0: Disable wake-up on WUC timeout event																											

## Firmware Timer Setup

The UHF Transceiver will wake up from the PHY\_SLEEP state at the rate set by the hardware WUC. A firmware timer, implemented by the on-chip processor, can be used to count the number of hardware wakeups and generate an interrupt to the host processor. This firmware timer is used in smart wake mode to determine the number of hardware wakeups. Thus, the UHF Transceiver can be used to handle the wake up timing of the host processor, reducing overall system power consumption.

To set up the firmware timer, the host processor must set the



NUMBER\_OF\_WAKEUPS\_IRQ\_THRESHOLD [15:0] (Address 0x104 and 0x105) to the desired value. This 16-bit value represents the number of times the device will wake up before it interrupts the host processor. At each wake up, the UHF Transceiver will increment the NUMBER\_OF\_WAKEUPS [15:0] (Address 0x103). If this value exceeds the value set by NUMBER\_OF\_WAKEUPS\_IRQ\_THRESHOLD [15:0] register then the NUMBER\_OF\_WAKEUPS [15:0] value will be cleared to 0. At this time, if the INTERRUPT\_NUM\_WAKEUPS bit in INTERRUPT\_MASK\_0 register (Address 0x100) is set then the device will assert the IRQ\_GP3 pin and enter the PHY\_OFF state.

## Downloadable Firmware Modules

The program RAM memory of the UHF Transceiver can be used to store firmware modules for the communications processor that provide extra functionality not already included in the ROM memory. The binary code for these firmware modules and detail on their functionality are available from ADI.

Three modules are briefly described here, namely:

- Image rejection calibration
- Reed-Solomon coding
- AES encryption and decryption

### Writing a module to program RAM

The sequence to write a firmware module to program RAM is as follows:

- Ensure that the UHF Transceiver is in PHY\_OFF.
- Issue the CMD\_RAM\_LOAD\_INIT command
- Write the module to program RAM using an SPI memory block write
- Issue the CMD\_RAM\_LOAD\_DONE command.
- Issue the CMD\_SYNC command.

The firmware module is now stored on program RAM.

### Downloadable Firmware Module: Image Rejection Calibration

The calibration system initially disables the UHF receiver, and an internal RF source is applied to the RF input at the image frequency. The algorithm then maximizes the receiver image rejection performance by iteratively minimizing the quadrature amplitude and phase errors in the polyphase filter.

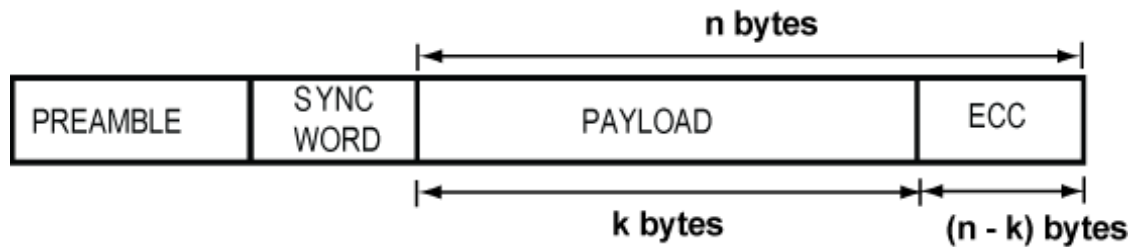
The calibration algorithm takes its initial estimates for quadrature phase correction (Address 0x118) and quadrature amplitude correction (Address 0x119) from BBRAM. After calibration, new optimum values of phase and gain are loaded back into these locations. These calibration values are maintained in BBRAM during sleep mode and are automatically reapplied from a wake-up event, which keeps the number of calibrations required to a minimum.

Depending on the initial values of quadrature amplitude and phase correction, the calibration algorithm can take approximately 20 ms to find the optimum image rejection performance. However, the calibration time can be significantly less than this when the seed values used for amplitude and phase corrections are close to optimum.

The image rejection performance is also dependent on temperature. To maintain optimum image rejection performance, a calibration should be activated whenever a temperature change of more than 10°C occurs. The UHF Transceiver on-chip temperature sensor can be used to determine when the temperature exceeds this limit.

## Downloadable Firmware Module: Reed Solomon Coding

This coding module uses Reed-Solomon block coding to detect and correct errors in the received packet. A transmit message of  $k$  bytes in length, is appended with an error check code (ECC) of length  $n-k$  bytes to give a total message length of  $n$  bytes, as shown in Figure 39.



**Figure 39: Packet Structure with appended Reed-Solomon Error Check Code (ECC)**

The receiver decodes the ECC to detect and correct up to  $t$  bytes in error in the received message where  $t = (n-k)/2$ . The firmware supports correction of up to five bytes in the  $n$  byte field. To correct  $t$  bytes in error, an ECC length of  $2t$  bytes is required, and the byte errors can be randomly distributed throughout the payload and ECC fields.

Reed Solomon coding exhibits excellent burst error correction capability and is commonly used to improve the robustness of a radio link in the presence of transient interference or due to rapid signal fading conditions that can corrupt sections of the message payload.

Reed Solomon coding is also capable of improving the receiver's sensitivity performance by several dB, where random errors tend to dominate under low SNR conditions and the receiver's packet error rate performance is limited by thermal noise.

The number of consecutive bit errors that can be 100% corrected is  $\{(t - 1) \times 8 + 1\}$ . Longer, random bit-error patterns, up to  $t$  bytes, can also be corrected if the error patterns start and end at byte boundaries.

The firmware also takes advantage of an on-chip hardware accelerator module to enhance throughput and minimize the latency of the Reed Solomon processing.

## Downloadable Firmware Module: AES Encryption and Decryption

The downloadable AES firmware module supports 128-bit block encryption and decryption with key sizes of 128 bits, 192 bits and 256 bits. Two modes are supported, ECB mode and CBC mode 1. ECB mode simply encrypts/decrypts on a 128 bit block by block with a single secret key as illustrated in Figure 40.

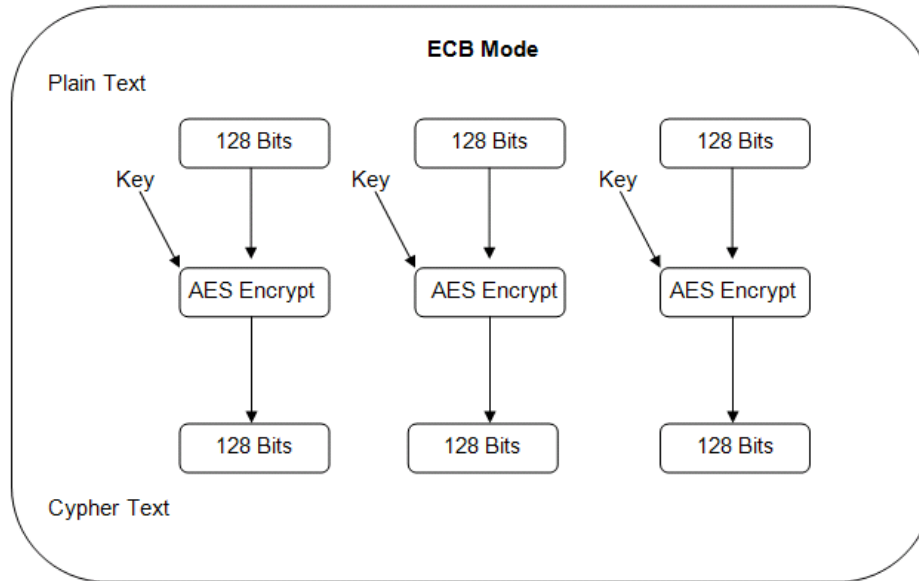


Figure 40: ECB Mode

CBC mode 1 encrypts after first adding (modulo 2) a 128 bit user supplied initialization vector. The resulting ciphertext is then used as the initialization vector for the next block and so forth as illustrated in Figure 41.

Decryption provides the inverse functionality. The firmware also takes advantage of an on-chip hardware accelerator module to enhance throughput and minimize the latency of the AES processing.

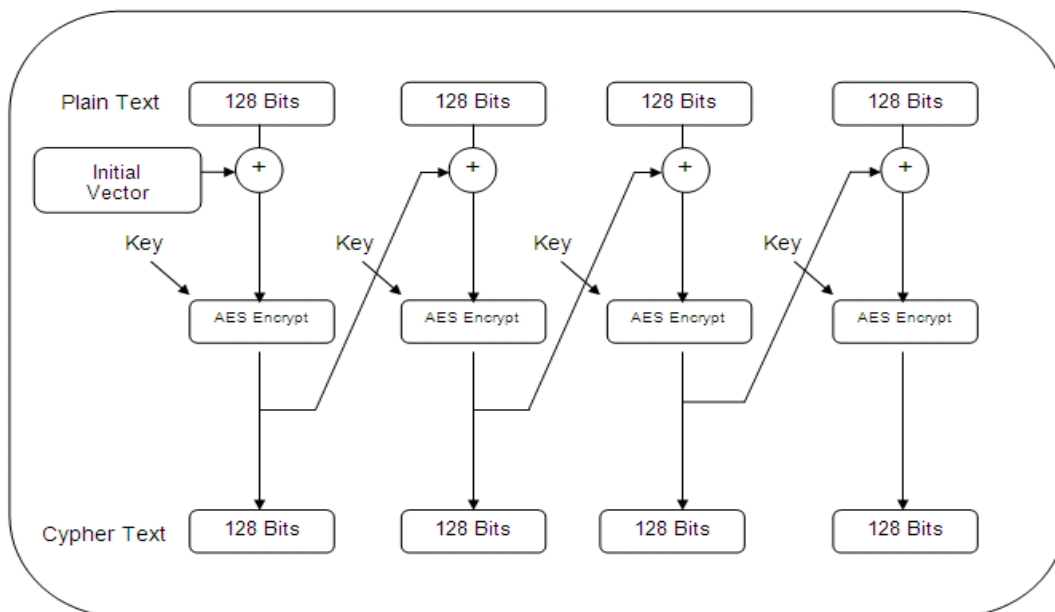


Figure 41: CBC Mode1

# Transceiver Radio Blocks

## RF Frequency Synthesizer

A fully integrated RF frequency synthesizer is used to generate both the transmit signal and receive local oscillator (LO) signal. The architecture of the frequency synthesizer is shown in Figure 42.

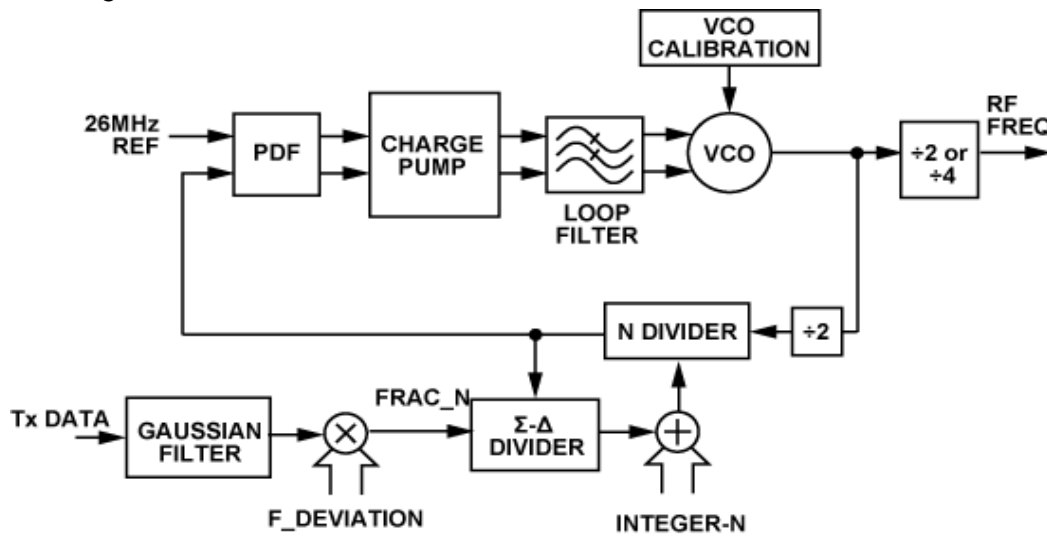


Figure 42: Synthesizer Architecture

The receiver uses a fractional-N frequency synthesizer to generate the mixer's LO for down conversion to the intermediate frequency (IF) of 200 kHz or 300 kHz. In transmit mode, a high resolution sigma-delta ( $\Sigma$ - $\Delta$ ) modulator is used to generate the required frequency deviations at the RF output when FSK data is transmitted. To reduce the occupied FSK bandwidth, the transmitted bit stream can be filtered using a digital Gaussian filter, which is enabled via the RADIO\_CFG\_9 register (Address 0x115). The Gaussian filter uses a bandwidth time (BT) of 0.5.

The VCO and the PLL loop filter of the UHF Transceiver are fully integrated. To reduce the effect of pulling of the VCO by the power-up of the PA and to minimize spurious emissions, the VCO operates at twice or four times the RF frequency. The VCO signal is then divided by 2 or 4, giving the required frequency for the transmitter and the required LO frequency for the receiver.

A high speed, fully automatic calibration scheme is used to ensure that the frequency and amplitude characteristic of the VCO are maintained over temperature, supply voltage, and process variations.

The calibration is automatically performed when the CMD\_PHY\_RX or CMD\_PHY\_TX command is issued. The calibration duration is 142  $\mu$ s, and if required, the CALIBRATION\_STATUS register (Address 0x339) can be polled to indicate the completion of the VCO self-calibration. After the VCO is calibrated, the frequency synthesizer settles to within  $\pm 5$  ppm of the target frequency in 56  $\mu$ s.

## Synthesizer Bandwidth

The synthesizer loop filter is fully integrated on chip and has a programmable bandwidth. The communications processor automatically sets the bandwidth of the synthesizer when the device enters the PHY\_TX or PHY\_RX state. On entering the PHY\_TX state the communications processor chooses the bandwidth based on the programmed modulation scheme (2FSK/GFSK or OOK) and the data rate. This ensures optimum modulation quality for each data rate. On entering the PHY\_RX state the communications processor sets a narrow bandwidth to ensure best receiver rejection. In all there are eight bandwidth configurations. Each synthesizer bandwidth setting is described in Table 104.

**Table 104: Automatic Synthesizer Bandwidth Selections**

Description	Data rate (kbps)	Closed loop synthesizer bandwidth (kHz)
Rx 2FSK/GFSK/MSK/GMSK	All	92
Tx 2FSK/GFSK/MSK/GMSK	1 – 49.5	130
Tx 2FSK/GFSK/MSK/GMSK	49.6 – 99.1	174
Tx 2FSK/GFSK/MSK/GMSK	99.2 – 129.5	174
Tx 2FSK/GFSK/MSK/GMSK	129.6 – 179.1	226
Tx 2FSK/GFSK/MSK/GMSK	179.2 – 239.9	305
Tx 2FSK/GFSK/MSK/GMSK	240 - 300	382
Tx OOK	All	185

## Synthesizer Settling

After the VCO calibration, a 56  $\mu$ s delay is required for synthesizer settling. This delay is fixed at 56 $\mu$ s by default and ensures that the synthesizer has fully settled when using any of the default synthesizer bandwidths. However in some cases it may be necessary to use a custom synthesizer settling delay. To use a custom delay set the bit CUSTOM\_TRX\_SYNTH\_LOCK\_TIME\_EN = 1 in the mode\_control register (Address 0x11A). The synthesizer settling delays for the PHY\_RX and PHY\_TX state transitions can be set independently in the RX\_SYNTH\_LOCK\_TIME register (Address 0x13E) and the TX\_SYNTH\_LOCK\_TIME register (Address 0x13F). The settling time can be set in the range 2 $\mu$ s to 512 $\mu$ s in steps of 2 $\mu$ s.

## Bypassing VCO Calibration

It is possible to bypass the VCO calibrations for ultra fast frequency hopping in transmit or receive. The calibration data for each RF channel should be stored in the host processor memory. The calibration data comprises of two values: the VCO band select value and the VCO amplitude level.

### Read and Store Calibration Data

- Go to PHY\_TX or PHY\_RX state without bypassing the VCO calibration.
- Read the following MCR registers and store the calibrated data in memory on the host processor:  
VCO\_BAND\_READBACK (Address 0x3DA)  
VCO\_AMPL\_READBACK (Address 0x3DB)

### Bypassing VCO Calibration on CMD\_PHY\_TX or CMD\_PHY\_RX

- Ensure the BBRAM is configured
- Set VCO\_OVRW\_EN = 0x3 (Address 0x3CD)
- Set VCO\_CAL\_CFG = 0x0F (Address 0x3D0)
- Set VCO\_BAND\_OVRW\_VAL = stored VCO\_BAND\_READBACK for that channel
- Set VCO\_AMPL\_OVRW\_VAL = stored VCO\_AMPL\_READBACK for that channel
- Set SYNTH\_CAL\_EN = 0 in the CALIBRATION\_CONTROL register. (Address 0x338)
- Set SYNTH\_CAL\_EN = 1 in the CALIBRATION\_CONTROL register. (Address 0x338)
- Issue CMD\_PHY\_TX or CMD\_PHY\_RX to go to PHY\_TX or PHY\_RX state without the VCO calibration.

## Modulation

The UHF transceiver supports binary frequency shift keying (2FSK), minimum shift keying (MSK), binary level Gaussian filtered 2FSK (GFSK), Gaussian filtered MSK (GMSK) and On-Off Keying (OOK). The desired transmit and receive modulation formats are set in the RADIO\_CFG\_9 register (Address 0x115).

When using 2FSK/GFSK/MSK/GMSK modulation, the frequency deviation can be set using the **FREQ\_DEVIATION** [11:0] parameter in the **RADIO\_CFG\_1** register (Address 0x10D) and **RADIO\_CFG\_2** register (Address 0x10E). The data rate can be set in the 1 kbps to 300 kbps range using the **DATA\_RATE** [11:0] parameter in the **RADIO\_CFG\_0** register (Address 0x10C) and **RADIO\_CFG\_1** register (Address 0x10D). For GFSK/GMSK modulation, the Gaussian filter uses a fixed bandwidth time (BT) product of 0.5.

When using OOK modulation, it is recommended to enable Manchester encoding (**MANCHESTER\_ENC** = 1, Address 0x11C). The data rate can be set in the 2.4 kbps to 19.2 kbps range (4.8 kcps to 38.4 kcps Manchester encoded) using the **DATA\_RATE** [11:0] parameter in the **RADIO\_CFG\_0** register (Address 0x10C) and **RADIO\_CFG\_1** register (Address 0x10D).

## Crystal Oscillator

A 26 MHz crystal oscillator operating in parallel mode must be connected between the XOSC26P and XOSC26N pins. Two parallel loading capacitors are required for oscillation at the correct frequency. Their values are dependent upon the crystal specification. They should be chosen to ensure that the shunt value of capacitance added to the PCB track capacitance and the input pin capacitance of the UHF Transceiver equals the specified load capacitance of the crystal, usually 10 pF to 20pF. Track capacitance values vary from 2 pF to 5 pF, depending on board layout. The total load capacitance is described by

$$C_{LOAD} = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2}} + \frac{C_{PIN}}{2} + C_{PCB}$$

where:

$C_{LOAD}$  is the total load capacitance.

$C_1$  and  $C_2$  are the external crystal load capacitors.

$C_{PIN}$  is the UHF Transceiver input capacitance of the XOSC26P and XOSC26N pins and is equal to 2.1pF.

$C_{PCB}$  is the PCB track capacitance.

When possible, choose capacitors that have a very low temperature coefficient to ensure stable frequency operation over all conditions.

The crystal frequency error can be corrected by means of an integrated digital tuning varactor. For a typical crystal load capacitance of 10 pF, a tuning range of –15 ppm to +11.25 ppm is available via programming of a 3-bit DAC. The 3-bit value should be written to XOSC\_CAP\_DAC in the OSC\_CONFIG register (Address 0x3D2).

Alternatively, any error in the RF frequency due to crystal error can be adjusted for by offsetting the RF channel frequency using the RF channel frequency setting in BBRAM memory.

## RF Output Stage

### Power Amplifier (PA)

The PA of the UHF Transceiver can be configured for single-ended or differential output operation using the PA\_SINGLE\_DIFF\_SEL bit in the RADIO\_CFG\_8 register (Address 0x114). The PA level is set by the PA\_LEVEL bit in the RADIO\_CFG\_8 register and has a range of 0 to 15. For finer control of the output power level, the PA\_LEVEL\_MCR register (Address 0x307) can be used. It offers more resolution with a setting range of 0 to 63. The relationship between the PA\_LEVEL and PA\_LEVEL\_MCR settings is given by

$$PA\_LEVEL\_MCR = 4 \times PA\_LEVEL + 3$$

The single-ended configuration can deliver 13.5 dBm output power. The differential PA can deliver 10 dBm output power and allows a straightforward interface to dipole antennae. The two PA configurations offer a Tx antenna diversity capability. Note that the two PAs cannot be enabled at the same time.



## Automatic PA Ramp

The UHF Transceiver has built-in up and down PA ramping for both single-ended and differential PA's. There are eight ramp rate settings, defined as a certain number of PA power level settings for each data bit period. The PA steps through each PA code setting, at a rate defined by the PA\_RAMP setting in the RADIO\_CFG\_8 register (Address 0x114) as outlined in Table 105 and illustrated in Figure 43.

The PA ramps to the level set by the PA\_LEVEL or PA\_LEVEL\_MCR settings. Enabling the PA ramp reduces spectral splatter and helps in meeting radio regulations which limit PA transient spurs. To ensure optimum performance an adequately long PA ramp rate is required based on the data rate and the PA output power setting. The PA\_RAMP setting should therefore be set such that:

$$\text{Ramp Rate}(\text{Codes} / \text{Bit}) < 2500 \times \frac{\text{PA\_LEVEL\_MCR}[5:0]}{\text{DATA\_RATE}[11:0]}$$

Where:

PA\_MCR\_LEVEL is related to the PA\_LEVEL setting by

$$\text{PA\_LEVEL\_MCR} = 4 \times \text{PA\_LEVEL} + 3.$$

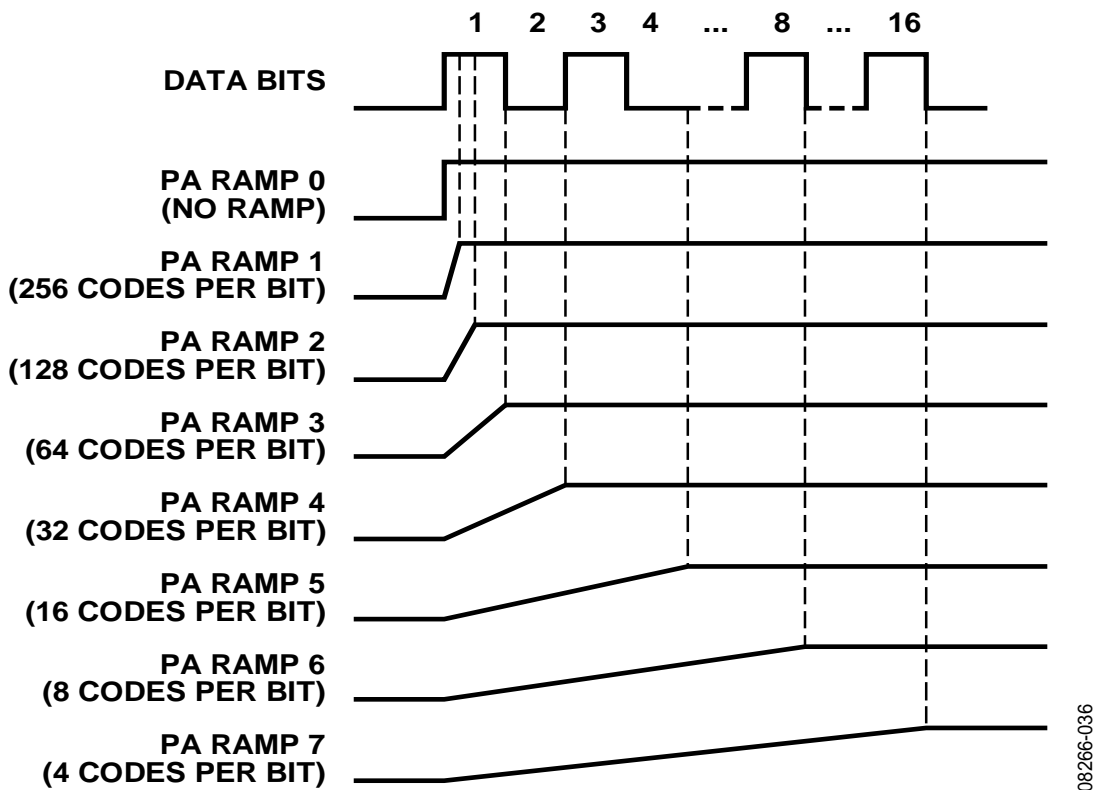


Figure 43: PA Ramp for Different PA\_RAMP Settings

Table 105: PA Ramp Rate Settings

Ramp Rate Setting	Ramp Time
000	No ramp
001	$\frac{1}{4}$ x Bit period
010	$\frac{1}{2}$ x Bit period
011	1 x Bit period
100	2 x Bit period
101	4 x Bit period
110	8 x Bit period
111	16 x Bit period

## PA/LNA Interface

The UHF transceiver supports both single-ended and differential PA outputs. Only one PA can be active at one time. The differential PA and LNA share the same pins, RFIO\_1P and RFIO\_1N, which facilitate a simpler antenna interface. The single-ended PA output is available on the RFO2 pin.

The RF interfaces are shown in Figure 44 when the single ended PA configuration is used.

To facilitate a simpler interface to a dipole antenna, the PA can be configured with a differential output. This differential signal appears on the LNA input pins, RFIO\_1P and RFIO\_1N. This is shown in Figure 45.

The differential PA configuration is enabled by setting `pa_single_diff_sel`.

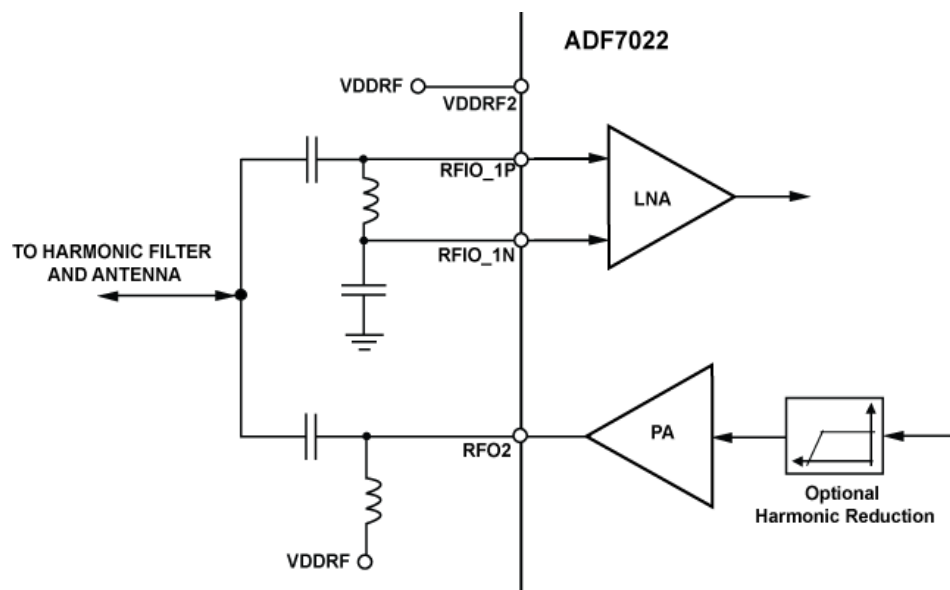


Figure 44: PA/LNA Interface with Single Ended PA Configuration

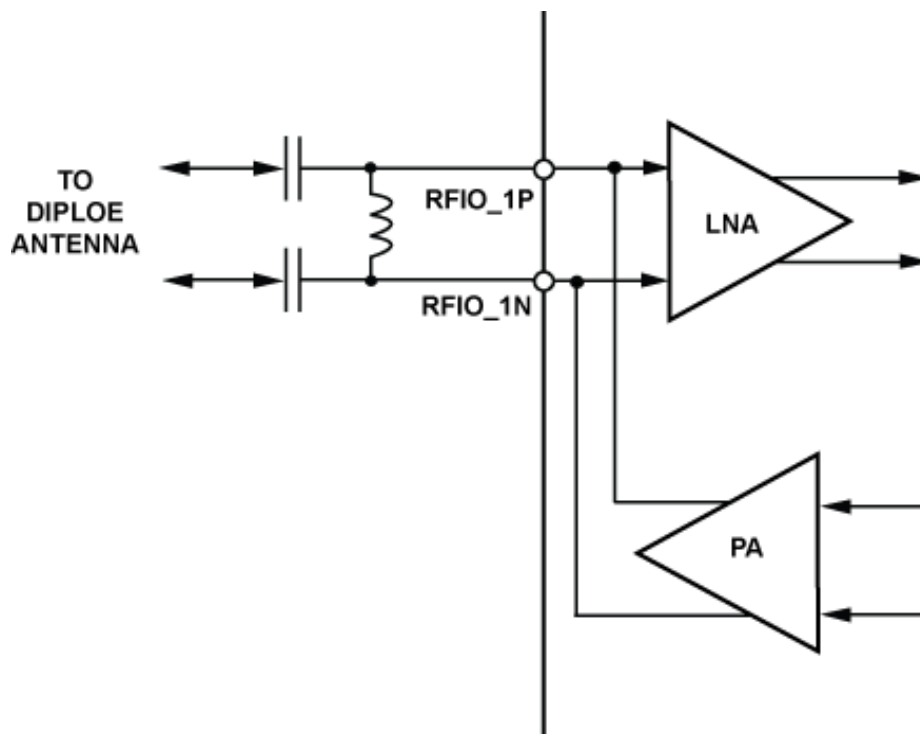


Figure 45: PA/LNA Interface with Differential PA Configuration

## Receive Channel Filter

The receiver IF filter is a fourth-order, active polyphase Butterworth filter. With programmable bandwidths of 100kHz, 150kHz, 200kHz and 300kHz. The fourth-order filter gives very good interference suppression of adjacent and neighbouring channels and also suppresses the image channel by approximately 36dB at a 100kHz IF bandwidth and an RF frequency of 868MHz or 915MHz.

For channel bandwidths of 100 kHz to 200 kHz, an IF frequency of 200 kHz is used, which results in an image frequency located 400 kHz below the wanted RF frequency. When the 300 kHz bandwidth is selected, an IF frequency of 300 kHz is used, and the image frequency is located at 600 kHz below the wanted frequency.

The bandwidth and center frequency of the IF filter are calibrated automatically after entering the PHY\_ON state if the BB\_CAL bit is set in the MODE\_CONTROL register (Address 0x11A). The filter calibration time takes 100  $\mu$ s.

The IF bandwidth is programmed by setting the IFBW field in the RADIO\_CFG\_9 register (Address 0x115). The filter's pass band is centered at an IF frequency of 200 kHz when bandwidths of 100 kHz to 200 kHz are used and centered at 300 kHz when an IF bandwidth of 300 kHz is used.

## Image Channel Rejection

The UHF transceiver is capable of providing improved image rejection performance by the use of a fully integrated image rejection calibration system, combined with a firmware

download that is written to the on-chip program RAM and is run by the communications processor.

## Automatic Gain Control (AGC)

AGC is enabled by default, and keeps the receiver gain at the correct level by selecting the LNA, mixer and filter gain settings based on the measured RSSI level. The LNA has three gain levels; the mixer has two gain levels, and the filter three gain levels. In all there are six AGC stages which are defined in Table 106.

**Table 106: AGC Gain Stages**

Gain Mode	LNA Gain (bits 4:3)	Mixer Gain (bit 2)	Filter Gain (bits 1:0)
6	Low (00)	Low (0)	Low (00)
5	Low (00)	Low (0)	Medium (01)
4	Low (00)	Low (0)	High (10)
3	Medium (01)	Low (0)	High (10)
2	High (10)	Low (0)	High (10)
1	High (10)	High (1)	High (10)

The AGC remains at each gain stage for a time defined by the AGC\_CLK\_DIVIDE register (Address 0x32F). The default value of AGC\_CLK\_DIVIDE = 0x28 gives an AGC delay of 25 $\mu$ s. When the RSSI is above the AGC\_HIGH\_THRESHOLD (Address 0x35F), the gain is reduced.

The AGC can be configured to remain active while in PHY\_RX or it can be locked on preamble detection. The AGC can also be set to manual mode in which case the host processor must set the LNA, filter and mixer gains by writing to the AGC\_MODE register (Address 0x35D). The AGC operation is set by the AGC\_LOCK\_MODE setting in the RADIO\_CFG\_7 register (Address 0x113) and is described in Table 107.

The LNA, filter and mixer gains can be readback through the AGC\_GAIN\_STATUS register (Address 0x360).

**Table 107: AGC Operation**

AGC_OPERATION (0x113)	Description
0	AGC is free running
1	AGC is disabled. Gains must be set manually
2	AGC is held at current gain level
3	AGC is locked on preamble detection

## RSSI

The RSSI is implemented as a successive compression, log amp architecture following the analog channel filtering.

The analog RSSI level is converted for user readback and for the digitally controlled AGC by an 8-bit SAR ADC.

The UHF transceiver has a total of four RSSI measurement functions that support a wide range of applications. These functions can be used to implement carrier sense (CS) or clear channel assessment (CCA). In packet mode, the RSSI is automatically recorded in MCR memory and is available for user readback after receipt of a packet. Table 108 details the four RSSI measurement methods.

Table 108. Summary of RSSI Measurement Methods					
RSSI Method	RSSI Type	Modulation	Available in Packet Mode	Available in Sport Mode	Description
1	Automatic end of packet RSSI	2FSK/GFSK/MSK/GMSK	Yes	No	Automatic RSSI measurement during reception of the postamble in packet mode. The RSSI result is available in the RSSI_READBACK register (Address 0x312).
2	CMD_GET_RSSI command from PHY_ON	2FSK/GFSK/MSK/GMSK	Yes	Yes	Automatic RSSI measurement from PHY_ON using CMD_GET_RSSI. The RSSI result is available in the RSSI_READBACK register (Address 0x312).
3	RSSI via ADC and AGC readback, FSK	2FSK/GFSK/MSK/GMSK	Yes	Yes	RSSI measurement based on the ADC and AGC gain readbacks. The host processor calculates RSSI in dBm.
4	RSSI via ADC and AGC readback, OOK	OOK	Yes	Yes	RSSI measurement based on the ADC and AGC gain readbacks. The host processor calculates RSSI in dBm.

### RSSI Method 1

When a valid packet is received in packet mode, the RSSI level during postamble is automatically loaded to the RSSI\_READBACK register (Address 0x312) by the communications processor. The RSSI\_READBACK register contains a twos complement value and can be converted to input power in dBm using the following formula:

$$\text{RSSI(dBm)} = \text{RSSI\_READBACK} - 107$$

To extend the linear range of RSSI measurement down to an input power of -110dBm, a cosine adjustment can be applied using the following formula:

$$\text{RSSI(dBm)} = \cos\left(\frac{8}{\text{RSSI\_READBACK}}\right) \times \text{RSSI\_READBACK} - 106$$

Where COS(X) is the cosine of Angle X (radians).

## RSSI Method 2

The CMD\_GET\_RSSI command can be used from the PHY\_ON state to read the RSSI. This RSSI measurement method uses additional low pass filtering, resulting in a more accurate RSSI reading. The RSSI result is loaded to the RSSI\_READBACK register (Address 0x312) by the communications processor. The RSSI\_READBACK register contains a twos complement value and can be converted to input power in dBm using the following formula

$$\text{RSSI (dBm)} = \text{RSSI\_READBACK} - 107.$$

## RSSI Method 3

This method supports the measurement of RSSI by the host processor at any time while in the PHY\_RX state. The receiver input power can be calculated using the following procedure:

- Set AGC to hold by setting the AGC\_MODE register (Address 0x35D) = 0x40 (only necessary if AGC has not been locked on the preamble or sync word).
- Read back the AGC gain settings (AGC\_GAIN\_STATUS register, Address 0x360).
- Read the ADC\_READBACK [7:0] value (Address 0x327 and Address 0x328).
- Re-enable the AGC by setting the AGC\_MODE register (Address 0x35D) = 0x00 (only necessary if AGC has not already been locked on the preamble or sync word).
- Calculate the RSSI in dBm as follows:

$$\text{RSSI (dBm)} = \left( \text{ADC\_READBACK}[7:0] \times \frac{2}{7} + \text{Gain\_Correction} \right) - 119$$

Where

Gain\_Correction is determined by the value of the AGC\_GAIN\_STATUS register (Address 0x360) as shown in Table 109.

**Table 109. Gain Mode Correction for 2FSK/GFSK/MSK/GMSK RSSI**

AGC_GAIN_STATUS (Address 0x360)	GAIN_CORRECTION
0x00	44
0x01	35
0x02	26
0x0A	17
0x12	10
0x16	0

To simplify the RSSI calculation, the following approximation can be used by the host processor:

$$\frac{2}{7} \approx \frac{1}{4} \left( 1 + \frac{1}{8} + \frac{1}{64} \right)$$

#### RSSI Method 4

This method is used to provide RSSI readback when using OOK demodulation in the PHY\_RX state. The receiver input power can be calculated using the following procedure:

1. Set AGC to hold by setting the AGC\_MODE register (Address 0x35D) = 0x40 (only necessary if AGC has not been locked on the preamble or sync word).
2. Read back the AGC gain settings (AGC\_GAIN\_STATUS register, Address 0x360).
3. Read the ADC\_READBACK [7:0] value (Address 0x327 and Address 0x328).
4. Re-enable the AGC by setting the AGC\_MODE register (Address 0x35D) = 0x00 (only necessary if AGC has not already been locked on the preamble or sync word).
5. Calculate the RSSI in dBm as follows:

$$\text{RSSI(dBm)} = \left( \text{ADC\_READBACK}[7:0] \times \frac{2}{7} + \text{Gain\_Correction} \right) - 110$$

Where:

Gain\_Correction is determined by the value of the AGC\_GAIN\_STATUS register (Address 0x360) as shown in Table 109.

**Table 110. Gain Mode Correction for OOK RSSI**

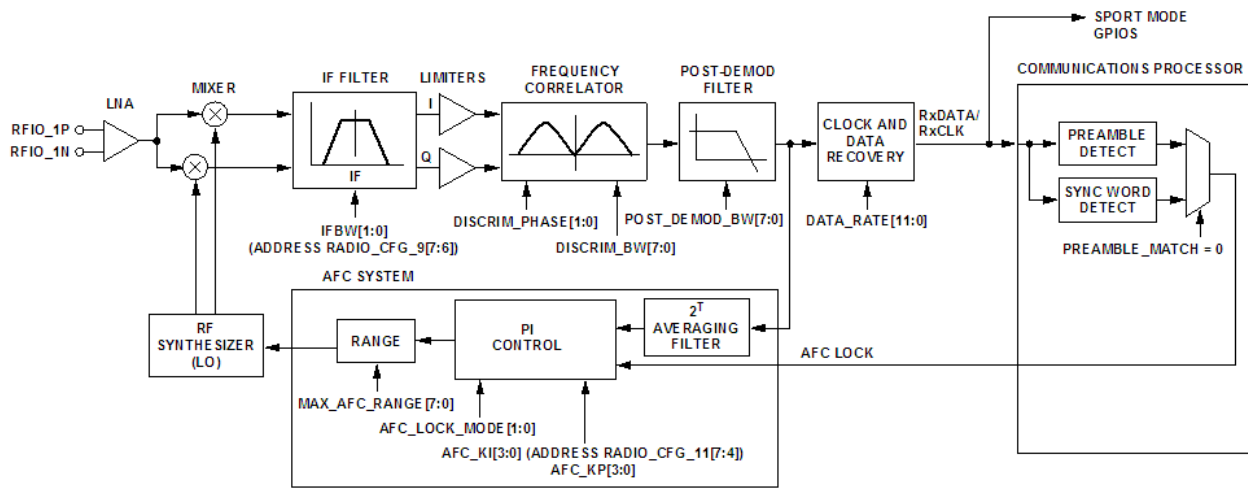
AGC_GAIN_STATUS(Address 0x360)	GAIN_CORRECTION
0x00	47
0x01	37
0x02	28
0x0A	19
0x12	10
0x16	0

To simplify the RSSI calculation, the following approximation can be used by the host processor:

$$\frac{2}{7} \approx \frac{1}{4} \left( 1 + \frac{1}{8} + \frac{1}{64} \right)$$

## 2FSK/GFSK/MSK/GMSK Demodulation

A correlator demodulator is used for 2FSK, MSK, GFSK and GMSK demodulation. The quadrature outputs of the IF filter are first limited and then fed to a digital frequency correlator that performs filtering and frequency discrimination of the 2FSK/GFSK/MSK/GMSK spectrum. Data is recovered by comparing the output levels from two correlators. The performance of this frequency discriminator approximates that of a matched filter detector, which is known to provide optimum detection in the presence of additive white Gaussian noise (AWGN). This method of 2FSK/GFSK/MSK/GMSK demodulation provides approximately 3 dB to 4 dB better sensitivity than a linear frequency discriminator. The 2FSK/GFSK/MSK/GMSK demodulator architecture is shown in Figure 46.



**Figure 46: FSK/GFSK Demodulation and AFC Architecture**

The UHF transceiver is set for 2FSK/GFSK/MSK/GMSK demodulation by setting DEMOD\_SCHEME = 0 in the RADIO\_CFG\_9 register (Address 0x115).

To optimize receiver sensitivity, the correlator bandwidth and phase must be optimized for the specific deviation frequency, data rate and maximum expected frequency error between the transmitter and receiver. The bandwidth and phase of the discriminator must be set using the DISCRIM\_BW bit in the RADIO\_CFG\_3 register (Address 0x10F) and DISCRIM\_PHASE [1:0] RADIO\_CFG\_5 register, (Address 0x112). The discriminator setup is performed in 3 steps:

### Step 1: Calculate the discriminator bandwidth coefficient K

The discriminator bandwidth coefficient K depends on the modulation index (MI) which is determined by:

$$MI = \frac{2 \times FSK\_dev}{data\_rate}$$

Where FSK\_dev is the 2FSK/GFSK frequency deviation in Hz, measured from the carrier



to the plus one symbol frequency (positive frequency deviation) or minus one symbol frequency (negative frequency deviation), and datarate is the data rate in bps.

The value of K is then determined by:

$$MI \geq 1, \text{ AFC off: } K = \text{Floor} \left[ \frac{IF\_Freq}{FSK\_Dev} \right]$$

$$MI < 1, \text{ AFC off: } K = \text{Floor} \left[ \frac{IF\_Freq}{\frac{Datarate}{2}} \right]$$

$$MI \geq 1, \text{ AFC on: } K = \text{Floor} \left[ \frac{IF\_Freq}{FSK\_Dev + Freq\_Error\_Max} \right]$$

$$MI < 1, \text{ AFC on: } K = \text{Floor} \left[ \frac{IF\_Freq}{\frac{Datarate}{2} + Freq\_Error\_Max} \right]$$

Where MI is the modulation index, K is the discriminator coefficient, Floor [] is a function to round down to the nearest integer, IF\_freq is the IF frequency in Hz (200 kHz or 300 kHz), FSK\_dev is the FSK/GFSK frequency deviation in Hz and Freq\_error\_max is the maximum expected frequency error, in Hz, between Tx and Rx.

### Step 2: Calculate the DISCRIM\_BW setting

The bandwidth setting of the discriminator is calculated based on the Discriminator Coefficient K and the IF frequency. The bandwidth is set using DISCRIM\_BW setting (Address 0x10F) which is calculated according to:

$$\text{discrim\_bw}[7:0] = \text{Round} \left[ \frac{K \times 3.25\text{MHz}}{IF\_freq} \right]$$

### Step 3: Calculate the DISCRIM\_PHASE setting

The phase setting of the discriminator is calculated based on the Discriminator Coefficient K as described in Table 111. The phase is set using the DISCRIM\_PHASE [1:0] value in the RADIO\_CFG\_6 register (Address 0x112).

Table 111: Setting DISCRIM_PHASE[1:0] Value Based on K			
K	K/2	(K+1)/2	discrim_phase [1:0]
Even	Odd	-	0
Odd		Even	1
Even	Even		2
Odd		Odd	3

## Automatic Frequency Control (AFC)

The UHF transceiver supports a real-time, internal, automatic frequency control loop. In receive, the control loop automatically monitors the frequency error during the packet preamble sequence and adjusts the receiver synthesizer local oscillator (LO) using an internal proportional integral (PI) control loop. The AFC frequency error measurement bandwidth is targeted specifically at the packet preamble sequence (DC Free). AFC is supported during 2FSK/GFSK/MSK/GMSK demodulation.

AFC can also be configured to lock on detection of the qualified preamble or on detection of the qualified sync word. To lock AFC on detection of the qualified preamble, set `ADC_LOCK_MODE = 3` (Address 0x116) and ensure that the preamble detection is enabled in the `PREAMBLE_MATCH` register (Address 0x11B). (AFC lock is released if the sync word is not detected immediately after the end of the preamble. In packet mode, if the qualified preamble is followed by a qualified sync word, the AFC lock is maintained for the duration of the packet. In sport mode, the AFC lock is released on transitioning back to the `PHY_ON` state or when a `CMD_PHY_RX` is issued while in the `PHY_RX` state.

To lock AFC on detection of the qualified sync word, set `AFC_LOCK_MODE = 3` and ensure that preamble detection is disabled in the `PREAMBLE_MATCH` register (Address 0x11B). If this mode is selected, consideration must be given to the selection of the sync word. The sync word should be dc free and have short run lengths yet low correlation with the preamble sequence. After lock on detection of the qualified sync word, the AFC lock is maintained for the duration of the packet. In sport mode, the AFC lock is released on transitioning back to the `PHY_ON` state or when `CMD_PHY_RX` is issued while in the `PHY_RX` state.

AFC is enabled by setting `AFC_LOCK_MODE` in the `radio_cfg_10` register (Address 0x116) as described in Table 112.

**Table 112: AFC\_LOCK\_MODE Register Settings**

<b>AFC_LOCK_MODE[1:0](Address 0x116)</b>	<b>AFC_LOCK_MODE</b>
00	Free Running: AFC is free running
01	Disabled: AFC is disabled
10	Hold AFC: AFC is paused
11	Lock: AFC locks on valid preamble detect

The bandwidth of the AFC loop can be controlled by the `AFC_KI` and `AFC_KP` parameters in the `RADIO_CFG_11` register (Address 0x117).

The maximum AFC pull in range is automatically set based on the programmed IF filter bandwidth (IFBW in the `RADIO_CFG_9` register, Address 0x115).

**Table 113: Maximum AFC Pull in Range**

IF Bandwidth	Max AFC pull in range
100kHz	±50kHz
150kHz	±75kHz
200kHz	±100kHz
300kHz	±150kHz

### AFC and Preamble Length

The AFC requires a certain number of the received preamble bits to correct the frequency error between the transmitter and the receiver. The number of preamble bits required depends on the datarate and on whether the AFC is locked on detection of a qualified preamble or locked on detection of a qualified sync word.

### AFC Readback

The frequency error between the received carrier and the Receiver local oscillator can be measured when AFC is enabled. The error value can be read from the `FREQUENCY_ERROR_READBACK` register (Address 0x372), where each LSB equates to 1 kHz. The value is a twos complement number. The `FREQUENCY_ERROR_READBACK` value is valid in `PHY_RX` after the AFC has been locked. The value is retained in the `FREQUENCY_ERROR_READBACK` register after recovering a packet and transitioning back to the `PHY_ON` state.

### Post-demodulator Filter

A second-order, digital low-pass filter removes excess noise from the demodulated bit stream at the output of the discriminator. The bandwidth of this post-demodulator filter is programmable and must be optimized for the user's data rate and received modulation type. If the bandwidth is set too narrow, performance degrades due to intersymbol interference (ISI). If the bandwidth is set too wide, excess noise degrades the performance of the receiver. For optimum performance, the post-demodulator filter bandwidth should be set close to 0.75 times the data rate (when using 2FSK/GFSK/MSK/GMSK modulation). The actual bandwidth of the post-demodulator filter is given by

$$\text{Post Demodulation filter bandwidth (kHz)} = \text{POST\_DEMOD\_BW} \times 2$$

Where

`POST_DEMOD_BW` is set in the `RADIO_CFG_4` register (Address 0x110)

## Clock Recovery

An oversampled digital Clock and Data Recovery (CDR) PLL is used to resynchronize the received bit stream to a local clock in all modulation modes. The maximum symbol rate tolerance of the CDR PLL is determined by the number of bit transitions in the transmitted bit stream.

For example, during reception of a 010101 preamble, the CDR achieves a maximum data rate tolerance of  $\pm 3.0\%$ . However, this tolerance is reduced during recovery of the remainder of the packet where symbol transitions may not be guaranteed to occur at regular intervals during the payload data. To maximize data rate tolerance of the receiver's CDR, 8b/10b encoding or Manchester encoding should be enabled, which guarantees a maximum number of contiguous bits in the transmitted bit stream. Data whitening can also be enabled on the UHF transceiver to break up long sequence of contiguous data bit patterns.

Using 2FSK/GFSK/MSK/GMSK modulation, it is also possible to tolerate uncoded payload data fields and payload data fields with long run length coding constraints if the data rate tolerance and packet length are both constrained. More details of CDR operation using uncoded packet formats are discussed in the analog devices application note AN-915.

The UHF transceiver CDR PLL has been optimized for fast acquisition of the recovered symbols during preamble and typically achieves bit synchronization within five symbol transitions of preamble.

## OOK Demodulation

The UHF transceiver can be set for OOK demodulation by setting `DEMOD_SCHEME = 2` in the `RADIO_CFG_9` register (Address 0x115). Manchester encoding should be used with OOK modulation to ensure optimum performance. OOK demodulation is performed using the receiver's RSSI signal in conjunction with a fully automatic threshold detection circuit, which extracts the optimum OOK threshold during preamble and maintains robust packet error performance over the full input power range. The bandwidth of the threshold detection circuit is set by the `AFC_KI` and `AFC_KP` parameters in the `RADIO_CFG_11` register (Address 0x117). The AGC loop bandwidth can be independently optimized for acquisition and tracking modes during OOK reception by setting `OOK_AGC_CLK_ACQ` and `OOK_AGC_CLK_TRK` (Address 0x35B), respectively. This demodulation scheme delivers high receiver saturation performance in OOK mode. The receiver also supports OOK modulation depths of up to 20 dB.

For optimum performance, the AGC and threshold detection circuit should be set to lock after preamble detection by setting `AGC_LOCK_MODE = 3` in the `RADIO_CFG_7` register (Address 0x113) and `AFC_LOCK_MODE = 3` in the `RADIO_CFG_10` register (Address 0x116).

The recommended post-demodulator filter bandwidth is 1.6 times the chip rate when using OOK demodulation. This can be configured via the `POST_DEMOD_BW` setting in the `RADIO_CFG_4` register (Address 0x110).

## Recommended Receiver Settings for 2FSK/GFSK/MSK/GMSK

To optimize the UHF Transceiver receiver performance and to ensure the lowest possible packet error rate, it is recommended to use the following configurations:

- Set the recommended AGC low and high thresholds and the AGC clock divide.
- Set the recommended AFC Ki and Kp parameters.
- Use a preamble length  $\geq$  the minimum recommended preamble length.
- When the AGC is configured to lock on the sync word at data rates greater than 200 kbps, it is recommended to set the sync word error tolerance to one bit.

The recommended settings for AGC, AFC, preamble length, and sync word are summarized in Table 115.

## Recommended AGC Settings

To optimize the receiver for robust packet error rate performance, when using minimum preamble length over the full input power range, it is recommended to overwrite the default AGC settings in the MCR memory. The recommended settings are as follows:

- AGC\_HIGH\_THRESHOLD (Address 0x35F) = 0x78
- AGC\_LOW\_THRESHOLD (Address 0x35E) = 0x46
- AGC\_CLOCK\_DIVIDE (Address 0x32F) = 0x0F or 0x19 (depends on the data rate; see Table 115.)

MCR memory is not retained in PHY\_SLEEP; therefore, to allow the use of these optimized AGC settings in low power mode applications, a static register fix can be used. An example static register fix to write to the AGC settings in MCR memory is shown in Table 114.

**Table 114. Example Static Register Fix for AGC Settings**

BBRAM Register	Data	Description
0x128 (STATIC_REG_FIX)	0x2B	Pointer to BBRAM Address 0x12B
0x12B	0x5E	MCR Address 0x35E
0x12C	0x46	Data to write to MCR Address 0x35E (sets AGC low threshold)
0x12D	0x5F	MCR Address 0x35F
0x12E	0x78	Data to write to MCR Address 0x35F (sets AGC high threshold)
0x12F	0x2F	MCR Address 0x32F
0x130	0x0F	Data to write to MCR Address 0x32F (sets AGC clock divide)
0x131	0x00	Ends static MCR register fixes

## Recommended AFC Settings

The bandwidth of the AFC loop is controlled by the AFC\_KI and AFC\_KP parameters in the RADIO\_CFG\_11 register (Address 0x117). To ensure optimum AFC accuracy while minimizing the AFC settling time (and thus the required preamble length), the AFC\_KI and AFC\_KP parameters should be set as outlined in Table 115.

## Recommended Preamble Length

When AFC is locked on preamble detection, the minimum preamble length is between 40 and 60 bits depending on the data rate. When AFC is set to lock on sync word detection, the minimum preamble length is between 14 and 32 bits, depending on the data rate. When AFC and preamble detection are disabled, the minimum preamble length is dependent on the AGC settling time and the CDR acquisition time and is between 8 and 24 bits, depending on the data rate. The required preamble length for various data rates and receiver configurations is summarized in Table 115.

## Recommended Sync Word Tolerance

At data rates greater than 200 kbps and when the AGC is configured to lock on the sync word, it is recommended to set the sync word error tolerance to one bit (SYNC\_ERROR\_TOL = 1). This prevents an AGC gain change during sync word reception causing a packet loss by allowing one bit error in the received sync word.

**Table 115: Summary of AGC, AFC, Preamble Length, and Sync Word Error Tolerance for 2FSK/GFSK/MSK/GMSK**

Data Rate (kbps)	Freq Deviation (kHz)	IF BW (kHz)	Setup <sup>1</sup>	AGC <sup>2</sup>			AFC <sup>3</sup>			Minimum Preamble Length (Bits) <sup>4</sup>	Sync Word Error Tolerance (Bits) <sup>5</sup>
				High Threshold	Low Threshold	Clock Divide	On/Off	Ki	Kp		
300	75	300	1	0x78	0x46	0x0F	On	7	3	64	0
			2	0x78	0x46	0x19	On	8	3	32	1
			3	0x78	0x46	0x19	Off			24	1
200	50	200	1	0x78	0x46	0x19	On	7	3	58	0
150	37.5	150	1	0x78	0x46	0x19	On	7	3	54	0
100	25	100	1	0x78	0x46	0x19	On	7	3	52	0
50	12.5	100	1	0x78	0x46	0x19	On	7	3	50	0
38.4	20	100	1	0x78	0x46	0x19	On	7	3	44	0
			2	0x78	0x46	0x19	On	7	3	14	0
			3	0x78	0x46	0x19	Off			8	0
9.6	10	100	1	0x78	0x46	0x19	Off			8	0
			1	0x78	0x46	0x19	On	7	3	46	0
1	10	100	1	0x78	0x46	0x19	Off			8	0
			1	0x78	0x46	0x19	On	7	3	40	0

<sup>1</sup> Setup 1: AFC and AGC are configured to lock on preamble detection by setting AFC\_LOCK\_MODE = 3 and AGC\_LOCK\_MODE = 3.

Setup 2: AFC and AGC are configured to lock on sync word detection by setting AFC\_LOCK\_MODE = 3, AGC\_LOCK\_MODE = 3, and PREAMBLE\_MATCH = 0.

Setup 3: AFC is disabled and AGC is configured to lock on sync word detection by setting AFC\_LOCK\_MODE = 1, AGC\_LOCK\_MODE = 3, and PREAMBLE\_MATCH = 0.

<sup>2</sup> The AGC high threshold is configured by writing to the AGC\_HIGH\_THRESHOLD register (Address 0x35F). The AGC low threshold is configured by writing to the AGC\_LOW\_THRESHOLD register (Address 0x35E). The AGC clock divide is configured by writing to the AGC\_CLOCK\_DIVIDE register (Address 0x32F).

<sup>3</sup> The AFC is enabled or disabled by writing to the AFC\_LOCK\_MODE setting in register RADIO\_CFG\_10 (Address 0x116). The AFC Ki and Kp parameters are configured by writing to the AFC\_KP and AFC\_KI settings in the RADIO\_CFG\_11 register (Address 0x117).

<sup>4</sup> The transmit preamble length (in bytes) is set by writing to the PREAMBLE\_LEN register (Address 0x11D).

<sup>5</sup> The sync word error tolerance (in bits) is set by writing to the SYNC\_ERROR\_TOL setting in the SYNC\_CONTROL register (Address 0x120).

## Recommended Receiver Settings for OOK

To ensure robust OOK reception, the AGC threshold detection, preamble length, and post-demodulator filter bandwidth are recommended to be set as detailed in Table 116.

**Table 116: Summary of Settings for AGC, AFC, and Preamble Length in OOK Demodulation**

Data Rate kbps	Chip Rate kcps	IF BW (kHz)	AGC <sup>1</sup>					Threshold Detection <sup>2</sup>			Minimum Preamble Length (Bits)	Post-Demodulator Bandwidth
			High Thres hold	Low Thres hold	AGC_LOCK_MODE	OOK_AGC_CLK_ACQ	OOK_AGC_CLK_TRK	AFC_KI	AFC_KP	AFC_LOCK_MODE		
2.4 to 19.2	4.8 to 38.4	100	0x69	0x2D	3	1	2	6	3	3	64	1.6 × chip rate

<sup>1</sup> The recommended values for the AGC high threshold (AGC\_HIGH\_THRESHOLD), OOK\_AGC\_CLK\_ACQ, and OOK\_AGC\_CLK\_TRK are the same as the default values and, therefore, do not need to be set by the host processor. The AGC low threshold is configured by writing to the AGC\_LOW\_THRESHOLD register (Address 0x35E). The AGC lock on preamble detection is configured by setting AGC\_LOCK\_MODE = 3 (in register RADIO\_CFG\_7, Address 0x113).

<sup>2</sup> The AFC\_KI and AFC\_KP parameters control the bandwidth of the threshold detection loop in OOK demodulation. They are configured by writing to the RADIO\_CFG\_11 register (Address 0x117). Setting AFC\_LOCK\_MODE = 3 configures the OOK threshold detection to lock on preamble detection.



# Peripheral Features

## Analog-to-Digital Converter

The UHF transceiver supports an integrated SAR ADC for digitization of analog signals that include the analog temperature sensor, the analog RSSI level, and an external analog input signal. The conversion time is typically 1  $\mu$ s. The result of the conversion can be read from register ADC\_READBACK\_HIGH (Address 0x327), and register ADC\_READBACK\_LOW (Address 0x328). The ADC readback is an 8-bit value.

The signal source for the ADC input is selected via the ADC\_CONFIG\_LOW register (Address 0x359). In the PHY\_RX state, the source is automatically set to the analog RSSI. The ADC is automatically enabled in the PHY\_RX state. In other radio states the host processor must enable the ADC by setting POWERDOWN\_RX (Address 0x324) = 0x10.

To perform an ADC readback, the following procedure should be followed:

- Read ADC\_READBACK\_HIGH. This initializes an ADC readback.
- Read ADC\_READBACK\_LOW. This returns the ADC\_READBACK [2:0] of the ADC sample
- Read ADC\_READBACK\_HIGH. This returns the ADC\_READBACK [7:3] of the ADC sample.

## Temperature Sensor

The integrated temperature sensor has an operating range between  $-40^{\circ}\text{C}$  and  $+85^{\circ}\text{C}$ .

To enable the temperature sensor in PHY\_OFF, PHY\_ON OR PHY\_TX, the following registers must be set:

- Set POWERDOWN\_RX (Address 0x324) = 0x10. This enables the ADC.
- Set POWERDOWN\_AUX (Address 0x325) = 0x02. This enables the temperature sensor.
- Set ADC\_CONFIG\_LOW (Address 0x359) = 0x08. This sets the ADC input to the temperature sensor.

The temperature is determined from the readback value using the following formula:

$$\text{Temperature } (^{\circ}\text{C}) = (\text{ADC\_READBACK [7:0]} \div 1.83) - 118.43 + \text{Correction Value}$$

The correction value can be determined by performing a readback at a single known temperature. When this correction is applied, the temperature sensor is accurate to  $\pm 14^{\circ}\text{C}$  over the full operating temperature range. Averaging a number of ADC readbacks can improve the accuracy of the temperature measurement. If an average of 10 readbacks is taken, the accuracy improves to  $\pm 4.4^{\circ}\text{C}$ .

## Test DAC

The test DAC allows the output of the post-demodulator filter to be viewed externally. It takes the 16-bit filter output and converts it to a high frequency, single-bit output using a second-order  $\Sigma$ - $\Delta$  converter. The output can be viewed on the GP0 pin. This signal, when filtered appropriately, can be used to:

- Monitor the signal at the post-demodulator filter output.
- Measure the demodulator output SNR.
- Construct an eye diagram of the received bit stream to measure the received signal quality.
- Implement analog FM demodulation.

To enable the test DAC the GPIO\_CONFIGURE setting (Address 0x3FA) should be set to 0xC9. The TEST\_DAC\_GAIN setting (Address 0x3FD) should be set to 0x00. The test DAC signal at the GP0 pin can be filtered with a 3-stage low-pass RC filter to reconstruct the demodulated signal.

## Transmit Test Modes

There are two transmit test modes which are enabled by setting the VAR\_TX\_MODE parameter (Address 0x00D in packet RAM memory) as described in Table 117. The VAR\_TX\_MODE setting should be set before entering the PHY\_TX state.

**Table 117:Transmit Test Modes**

VAR_TX_MODE	Mode
0	Default. No transmit test mode
1	Reserved
2	Transmit preamble continuously
3	Transmit carrier continuously
4 to 255	Reserved

## Silicon Revision Readback

The product code and silicon revision code can be read from the packet RAM memory as described in Table 118. The values of the product code and silicon revision code are valid only on power-up or wake-up from the PHY\_SLEEP state because the communications processor overwrites these values on transitioning from the PHY\_ON state.

**Table 118. Product Code and Silicon Revision Code**

Packet Ram Location	Description
0x001	Product code, most significant byte = 0x70
0x002	Product code, least significant byte = 0x23
0x003	Silicon revision code, most significant byte
0x004	Silicon revision code least significant byte

# BBRAM Register Maps

Table 119: Battery Backup Memory (BBRAM)

Address (Hex)	Register	Retained in PHY_SLEEP	Access	Group
0x100	INTERRUPT_MASK_0	Yes	R/W	MAC
0x101	INTERRUPT_MASK_1	Yes	R/W	MAC
0x102	NUMBER_OF_WAKEUPS_0	Yes	R/W	MAC
0x103	NUMBER_OF_WAKEUPS_1	Yes	R/W	MAC
0x104	NUMBER_OF_WAKEUPS_IRQ_THRESHOLD_0	Yes	R/W	MAC
0x105	NUMBER_OF_WAKEUPS_IRQ_THRESHOLD_1	Yes	R/W	MAC
0x106	RX_DWELL_TIME	Yes	R/W	MAC
0x107	PARMTIME_DIVIDER	Yes	R/W	MAC
0x108	SWM_RSSI_THRESH	Yes	R/W	PHY
0x109	CHANNEL_FREQ_0	Yes	R/W	PHY
0x10A	CHANNEL_FREQ_1	Yes	R/W	PHY
0x10B	CHANNEL_FREQ_2	Yes	R/W	PHY
0x10C	RADIO_CFG_0	Yes	R/W	PHY
0x10D	RADIO_CFG_1	Yes	R/W	PHY
0x10E	RADIO_CFG_2	Yes	R/W	PHY
0x10F	RADIO_CFG_3	Yes	R/W	PHY
0x110	RADIO_CFG_4	Yes	R/W	PHY
0x111	RADIO_CFG_5	Yes	R/W	PHY
0x112	RADIO_CFG_6	Yes	R/W	PHY
0x113	RADIO_CFG_7	Yes	R/W	PHY
0x114	RADIO_CFG_8	Yes	R/W	PHY
0x115	RADIO_CFG_9	Yes	R/W	PHY
0x116	RADIO_CFG_10	Yes	R/W	PHY
0x117	RADIO_CFG_11	Yes	R/W	PHY
0x118	IMAGE_REJECT_CAL_PHASE	Yes	R/W	PHY
0x119	IMAGE_REJECT_CAL_AMPLITUDE	Yes	R/W	PHY
0x11A	MODE_CONTROL	Yes	R/W	PHY
0x11B	PREAMBLE_MATCH	Yes	R/W	Packet
0x11C	SYMBOL_MODE	Yes	R/W	Packet
0x11D	PREAMBLE_LEN	Yes	R/W	Packet
0x11E	CRC_POLY_0	Yes	R/W	Packet
0x11F	CRC_POLY_1	Yes	R/W	Packet
0x120	SYNC_CONTROL	Yes	R/W	Packet
0x121	SYNC_BYTE_0	Yes	R/W	Packet
0x122	SYNC_BYTE_1	Yes	R/W	Packet
0x123	SYNC_BYTE_2	Yes	R/W	Packet
0x124	TX_BASE_ADR	Yes	R/W	Packet
0x125	RX_BASE_ADR	Yes	R/W	Packet
0x126	PACKET_LENGTH_CONTROL	Yes	R/W	Packet
0x127	PACKET_LENGTH_MAX	Yes	R/W	Packet
0x128	STATIC_REG_FIX	Yes	R/W	PHY
0x129	ADDRESS_MATCH_OFFSET	Yes	R/W	Packet
0x12A	Address Length	Yes	R/W	Packet
0x12Bto 0x13D	Address filtering	Yes	R/W	Packet
0x13E	RX_SYNTH_LOCK_TIME	Yes	R/W	PHY
0x13F	TX_SYNTH_LOCK_TIME	Yes	R/W	PHY

# Modem Configuration Register Map

All Modem Configuration registers return to their default value in PHY\_SLEEP.

**Table 120. Modem Configuration Memory (MCR)**

Address (Hex)	Register	Retained in PHY_SLEEP	Access
0x307	PA_LEVEL_MCR	No	R/W
0x30C	WUC_CONFIG_HIGH	No	W
0x30D	WUC_CONFIG_LOW	No	W
0x30E	WUC_VALUE_HIGH	No	W
0x30F	WUC_VALUE_LOW	No	W
0x310	WUC_FLAG_RESET	No	R/W
0x311	WUC_STATUS	No	R
0x312	RSSI_READBACK	No	R
0x315	MAX_AFC_RANGE	No	R/W
0x319	IMAGE_REJECT_CAL_CONFIG	No	R/W
0x322	CHIP_SHUTDOWN	No	R/W
0x324	POWERDOWN_RX	No	R/W
0x325	POWERDOWN_AUX	No	R/W
0x327	ADC_READBACK_HIGH	No	R
0x328	ADC_READBACK_LOW	No	R
0x32D	BATTERY_MONITOR_THRESHOLD_VOLTAGE	No	R/W
0x32E	EXT_UC_CLK_DIVIDE	No	R/W
0x32F	AGC_CLK_DIVIDE	No	R/W
0x336	INTERRUPT_SOURCE_0	No	R/W
0x337	INTERRUPT_SOURCE_1	No	R/W
0x338	CALIBRATION_CONTROL	No	R/W
0x339	CALIBRATION_STATUS	No	R
0x345	RXBB_CAL_CALWRD_READBACK	No	R
0x346	RXBB_CAL_CALWRD_OVERWRITE	No	RW
0x359	ADC_CONFIG_LOW	No	R/W
0x35A	ADC_CONFIG_HIGH	No	R/W
0x35B	AGC_OOK_CONTROL	No	R/W
0x35C	AGC_CONFIG	No	R/W
0x35D	AGC_MODE	No	R/W
0x35E	AGC_LOW_THRESHOLD	No	R/W
0x35F	AGC_HIGH_THRESHOLD	No	R/W
0x360	AGC_GAIN_STATUS	No	R
0x372	FREQUENCY_ERROR_READBACK	No	R
0x3CB	VCO_BAND_OVRW_VAL	No	R/W
0x3CC	VCO_AMPL_OVRW_VAL	No	R/W
0x3CD	VCO_OVRW_EN	No	R/W
0x3D0	VCO_CAL_CFG	No	R/W
0x3D2	OSC_CONFIG	No	R/W
0x3DA	VCO_BAND_READBACK	No	R
0x3DB	VCO_AMPL_READBACK	No	R
0x3F8	ANALOG_TEST_BUS	No	R/W
0x3F9	RSSI_TSTMUX_SEL	No	R/W
0x3FA	GPIO_CONFIGURE	No	R/W
0x3FD	TEST_DAC_GAIN	No	R/W

**Table 121. Packet RAM Memory**

Address	Register	Access
0x000	VAR_COMMAND	R/W
0x001 <sup>1</sup>	Product code, most significant byte = 0x70	R
0x002 <sup>1</sup>	Product code, least significant byte = 0x23	R
0x003 <sup>1</sup>	Silicon revision code, most significant byte	R
0x004 <sup>1</sup>	Silicon revision code, least significant byte	R
0x005 to 0x00C	Reserved	R
0x00D	VAR_TX_MODE	R/W
0x00E to 0x00F	Reserved	R

---

<sup>1</sup> Only valid on power-up or wake-up from the PHY\_SLEEP state because the communications processor overwrites these values on exit from the PHY\_ON state.

## BBRAM Register Description

**Table 122: 0x100 INTERRUPT\_MASK\_0**

Bits	Name	Description
7	INTERRUPT_NUM_WAKEUPS	Enable the interrupt for when the NUMBER_OF_WAKEUPS[15:0] reaches the NUMBER_OF_WAKEUPS_IRQ_THRESHOLD[15:0] of the firmware timer 1: Enabled 0: Disabled
6	INTERRUPT_SWM_RSSI_DET	Interrupt indicating that the RSSI threshold has been exceeded (Smart Wake Mode) 1: Enabled 0: Disabled
5	INTERRUPT_AES_DONE	Interrupt when an AES encryption or decryption command is completed 1: Enabled 0: Disabled
4	INTERRUPT_TX_EOF	Interrupt when a packet has finished transmitting 1: Enabled 0: Disabled
3	INTERRUPT_ADDRESS_MATCH	Interrupt when a received packet has a valid address match 1: Enabled 0: Disabled
2	INTERRUPT_CRC_CORRECT	Interrupt when a received packet has the correct CRC. 1: Enabled 0: Disabled
1	INTERRUPT_SYNC_DETECT	Interrupt when a qualified sync word has been detected in the received packet 1: Enabled 0: Disabled
0	INTERRUPT_PREMABLE_DETECT	Interrupt when a qualified preamble has been detected in the received packet. 1: Enabled 0: Disabled

**Table 123: 0x101 INTERRUPT\_MASK\_1**

Bits	Name	Description
7	BATTERY_ALARM	Interrupt when the battery voltage has dropped below the threshold value (BATTERY_MONITOR_THRESHOLD_VOLTAGE, Address 0x32D) 1: Enabled 0: disabled
6	CMD_READY	Interrupt when the communications processor is ready to load a new command; mirrors the CMD_READY bit of the status word 1: Enabled 0: disabled
5	Reserved	Reserved
4	WUC_TIMEOUT	Interrupt when the WUC has timed out 1: Enabled 0: disabled
3 to 2	Reserved	Reserved
1	SPI_READY	Interrupt when the SPI is ready for access 1: Enabled 0: disabled
0	CMD_FINISHED	Interrupt when the communications processor has finished performing a command 1: Enabled 0: disabled

**Table 124: 0x102 NUMBER\_OF\_WAKEUPS\_0**

Bits	Name	Description
7 to 0	NUMBER_OF_WAKEUPS[7:0]	Bits [7:0] of [15:0] of an internal 16-bit count of the number of wake ups (WUC timeouts) the device has gone through. It can be initialized to 0x0000.

**Table 125: 0x103 NUMBER\_OF\_WAKEUPS\_1**

Bits	Name	Description
7 to 0	NUMBER_OF_WAKEUPS[15:8]	Bits [15:8] of [15:0] of an internal 16-bit count of the number of WUC wake ups the device has gone through. Initialised to 0x0000.

**Table 126: 0x104 NUMBER\_OF\_WAKEUPS\_IRQ\_THRESHOLD\_0**

Bits	Name	Description
7 to 0	NUMBER_OF_WAKEUPS_IRQ_THRESHOLD[7:0]	Bits [7:0] of [15:0]. This is the threshold for the number of wakeups (WUC timeouts). It is a 16-bit count threshold that is compared against the NUMBER_OF_WAKEUPS parameter. When this threshold is exceeded the device wakes up into the state PHY_OFF and optionally generates INTERRUPT_NUM_WAKEUPS.

**Table 127: 0x105 NUMBER\_OF\_WAKEUPS\_IRQ\_THRESHOLD\_1**

Bits	Name	Description
7 to 0	NUMBER_OF_WAKEUPS_IRQ_THRESHOLD[15:8]	Bits [15:8] of [15:0]. It is a 16-bit count threshold that is compared against the NUMBER_OF_WAKEUPS parameter. When this threshold is exceeded the device wakes up into the state PHY_OFF and optionally generates INTERRUPT_NUM_WAKEUPS..

**Table 128: 0x106 RX\_DWELL\_TIME**

Bits	Name	Description
7 to 0	RX DWELL TIME	When the WUC is used and SWM is enabled, the radio powers up and enables the receiver on the channel defined in the BBRAM and listens for this period of time. If no preamble pattern is detected in this period, the device goes back to sleep. Receive Dwell Time (s) = $RX\_DWELL\_TIME \times 6.5MHz / (128 \times PARMTIME\_DIVIDER)$ .

**Table 129: 0x107 PARMTIME\_DIVIDER**

Bits	Name	Description
7 to 0	PARMTIME_DIVIDER	Units of time used to define the RX_DWELL_TIME time period. Timer Tick Rate= $(128 \times PARMTIME\_DIVIDER) / 6.5MHz$ A value of 0x33 will give a clock of 995.7Hz or a period of 1.004ms

**Table 130: 0x108 SWM\_RSSI\_THRESH**

Bits	Name	Description
7 to 0	SWM_RSSI_THRESH	This sets the RSSI threshold when in Smart Wake Mode with RSSI detection enabled. Threshold (dBm) = $SWM\_RSSI\_THRESH - 107$

**Table 131: 0x109 CHANNEL\_FREQ\_0**

Bits	Name	Description
7 to 0	CHANNEL_FREQ_0 [7:0]	The RF channel frequency bits [7:0] in Hz is set according to:  $Frequency(Hz) = F_{PFD} \times \frac{channel\_Freq[23:0]}{2^{16}}$  where $F_{PFD}$ is the PFD frequency and is equal to 26MHz.



**Table 132: 0x10A CHANNEL\_FREQ\_1**

Bits	Name	Description
7 to 0	CHANNEL_FREQ [15:8]	The RF channel frequency bits [15:8]

**Table 133: 0x10B CHANNEL\_FREQ\_2**

Bits	Name	Description
7 to 0	CHANNEL_FREQ[23:16]	The RF channel frequency bits [23:16]

**Table 134: 0x10C RADIO\_CFG\_0**

Bits	Name	Description
7 to 0	DATA_RATE[7:0]	Datarate LSB The datarate in bps is set according to: $\text{DataRate (bps)} = \text{DATA\_RATE}[11:0] \times 100$

**Table 135: 0x10D RADIO\_CFG\_1**

Bits	Name	Description
7 to 4	FREQUENCY_DEVIATION [11:8]	Frequency_deviation MSB
3 to 0	DATA_RATE [11:8]	Datarate MSB

**Table 136: 0x10E RADIO\_CFG\_2**

Bits	Name	Description
7 to 0	FREQ_DEVIATION[7:0]	The binary level 2FSK/GFSK/MSK/GMSK frequency deviation in Hz (defined as frequency difference between carrier frequency and 1/0 tones) is set according to: $\text{Frequency Deviation (Hz)} = \text{FREQ\_DEVIATION}[11:0] \times 100$

Table 137: 0x10F RADIO\_CFG\_3

Bits	Name	Description
7 to 0	DISCRIM_BW[7:0]	<p>The DISCRIM_BW value sets the bandwidth of the correlator demodulator.</p> <p>Step 1: Calculate the discriminator bandwidth coefficient K.</p> <p>The discriminator bandwidth coefficient K depends on the modulation index (MI) which is determined by:</p> $MI = \frac{2 \times FSK\_dev}{datarate}$ <p>Where FSK_dev is the 2FSK/GFSK/MSK/GMSK frequency deviation in Hz, measured from the carrier to the +1 (positive frequency deviation) or -1 negative frequency deviation, and datarate is the data rate in bps.</p> <p>The value of K is then determined by:</p> <p><i>MI ≥ 1, AFC off:</i></p> $K = \text{Floor} \left[ \frac{IF\_freq}{FSK\_dev} \right]$ <p><i>MI &lt; 1, AFC Off:</i></p> $K = \text{Floor} \left[ \frac{IF\_freq}{\frac{datarate}{2}} \right]$ <p><i>MI ≥ 1, AFC on:</i></p> $K = \text{Floor} \left[ \frac{IF\_freq}{FSK\_dev + Freq\_error\_max} \right]$ <p><i>M &lt; 1, AFC on:</i></p> $K = \text{Floor} \left[ \frac{IF\_freq}{\frac{datarate}{2} + Freq\_error\_max} \right]$ <p>Where MI is the modulation index, K is the discriminator coefficient, Floor[] is a function to round down to the nearest integer, IF_freq is the IF frequency in Hz (200kHz or 300kHz), FSK_dev is the 2FSK/GFSK/MSK/GMSK frequency deviation in Hz and Freq_error_max is the maximum expected frequency error, in Hz, between Tx and Rx.</p> <p>Step 2: Calculate the DISCRIM_BW setting</p> <p>The bandwidth setting of the discriminator is calculated based on the discriminator coefficient K and the IF frequency. The bandwidth is set using the DISCRIM_BW setting (Address 0x10F) which is calculated according to:</p> $discrim\_bw[7:0] = \text{Round} \left[ \frac{K \times 3.25\text{MHz}}{IF\_freq} \right]$

Table 138: 0x110 RADIO\_CFG\_4

Bits	Name	Description
7 to 0	POST_DEMOD_BW[7:0]	<p>For optimum performance, the post-demodulator filter bandwidth should be set close to 0.75 times the data rate. The actual bandwidth of the post-demod-ulator filter is given by</p> $\text{Post-Demodulator Filter Bandwidth (kHz)} = \text{POST\_DEMOD\_BW} \times 2$ <p>The range of POST_DEMOD_BW is 1 to 255.</p>

Table 139: 0x111 RADIO\_CFG\_5

Bits	Name	Description
7 to 0	Reserved	Reserved, Set to zero

Table 140: 0x112 RADIO\_CFG\_6

Bits	Name	Description																				
7 to 2	SYNTH_LUT_CONFIG_0	If SYNTH_LUT_CONTROL (Address 0x113) = 0 or 2, set SYNTH_LUT_CONFIG_0 = 0. If SYNTH_LUT_CONTROL = 1 or 3, this setting allows the receiver PLL loop bandwidth to be changed to optimize the receiver local oscillator phase noise.																				
1 to 0	DISCRIM_PHASE [1:0]	<div>The phase setting of the demodulator is calculated based on the discriminator coefficient K as described below.</div> <div>See the 2FSK/GFSK/MSK/GMSK Demodulation section for the steps required to set the DISCRIM_PHASE value.</div> <table><tr><th>K</th><th>K/2</th><th>(K+1)/2</th><th>discrim_phase[1:0]</th></tr><tr><td>Even</td><td>Odd</td><td></td><td>0</td></tr><tr><td>Odd</td><td></td><td>Even</td><td>1</td></tr><tr><td>Even</td><td>Even</td><td></td><td>2</td></tr><tr><td>Odd</td><td></td><td>Odd</td><td>3</td></tr></table>	K	K/2	(K+1)/2	discrim_phase[1:0]	Even	Odd		0	Odd		Even	1	Even	Even		2	Odd		Odd	3
K	K/2	(K+1)/2	discrim_phase[1:0]																			
Even	Odd		0																			
Odd		Even	1																			
Even	Even		2																			
Odd		Odd	3																			

Table 141: 0x113 RADIO\_CFG\_7

Bits	Name	Description									
7 to 6	AGC_LOCK_MODE	AGC_LOCK_MODE[1:0] 00:free running 01>manual 10:hold 11:lock after preamble/ sync word (only locks on sync word if PREAMBLE_MATCH =0)									
5 to 4	SYNTH_LUT_CONTROL	By default, the synthesizer loop bandwidth is automatically selected from look up tables (LUT) in ROM memory. A narrow bandwidth is selected in receive to ensure optimum interference rejection, whereas in transmit the bandwidth is selected based on the data rate and modulation settings. For the majority of applications, these automatically selected PLL loop bandwidths are optimum. However, in some applications it may be necessary to use custom transmit or receive bandwidths, in which case, various options exist.									
		SYNTH_LUT_CONTROL	Description	0	Use pre-defined transmit and receive LUTs. The LUTs are automatically selected from ROM memory on transitioning into the PHY_TX or PHY_RX state.	1	Use custom receive LUT based on SYNTH_LUT_CONFIG_0 and SYNTH_LUT_CONFIG_1. In transmit the pre-defined LUT in ROM is used.	2	Use a custom transmit LUT. The custom transmit LUT needs to be written to packet RAM locations 0x10 to 0x18. In receive the pre-defined LUT in ROM is used.	3	Use a custom receive LUT based on SYNTH_LUT_CONFIG_0 and SYNTH_LUT_CONFIG_1 and use a custom transmit LUT. The custom transmit LUT needs to be written to packet RAM locations 0x10 to 0x18.
		SYNTH_LUT_CONTROL	Description								
		0	Use pre-defined transmit and receive LUTs. The LUTs are automatically selected from ROM memory on transitioning into the PHY_TX or PHY_RX state.								
		1	Use custom receive LUT based on SYNTH_LUT_CONFIG_0 and SYNTH_LUT_CONFIG_1. In transmit the pre-defined LUT in ROM is used.								
		2	Use a custom transmit LUT. The custom transmit LUT needs to be written to packet RAM locations 0x10 to 0x18. In receive the pre-defined LUT in ROM is used.								
3	Use a custom receive LUT based on SYNTH_LUT_CONFIG_0 and SYNTH_LUT_CONFIG_1 and use a custom transmit LUT. The custom transmit LUT needs to be written to packet RAM locations 0x10 to 0x18.										
As packet RAM memory is lost in the PHY_SLEEP state the custom LUT for transmit must be re-loaded to packet RAM after waking from PHY_SLEEP.											
3 to 0	SYNTH_LUT_CONFIG_1	If SYNTH_LUT_CONTROL = 0 or 2, set SYNTH_LUT_CONFIG_0 to 0. If SYNTH_LUT_CONTROL = 1 or 3, this setting allows the receiver PLL loop bandwidth to be changed to optimize the receiver local oscillator phase noise.									

Table 142: 0x114 RADIO\_CFG\_8

Bits	Name	Description																		
7	PA_SINGLE_DIFF_SEL	Select the type of PA used:																		
		<table><tr><th>PA_SINGLE_DIFF_SEL</th><th>PA</th></tr><tr><td>0</td><td>Single ended PA enabled</td></tr><tr><td>1</td><td>Differential PA enabled</td></tr></table>	PA_SINGLE_DIFF_SEL	PA	0	Single ended PA enabled	1	Differential PA enabled												
		PA_SINGLE_DIFF_SEL	PA																	
		0	Single ended PA enabled																	
1	Differential PA enabled																			
6 to 3	PA_LEVEL	Sets the PA power:A value of zero sets the minimum RF output power, and a value of 15 sets the maximum PA output power. The PA level can also be set with finer resolution using the PA_LEVEL_MCR setting (Address 0x307) the PA_Level setting is related to the PA_LEVEL_MCR setting by PA_LEVEL_MCR = 4 x PA_LEVEL + 3.																		
		<table><tr><th>PA_POWER</th><th>PA level</th></tr><tr><td>0000</td><td>setting 3</td></tr><tr><td>0001</td><td>setting 7</td></tr><tr><td>0010</td><td>setting 11</td></tr><tr><td>:</td><td>:</td></tr><tr><td>:</td><td>:</td></tr><tr><td>1111</td><td>setting 63</td></tr></table>	PA_POWER	PA level	0000	setting 3	0001	setting 7	0010	setting 11	:	:	:	:	1111	setting 63				
		PA_POWER	PA level																	
		0000	setting 3																	
		0001	setting 7																	
		0010	setting 11																	
		:	:																	
		:	:																	
		1111	setting 63																	
2 to 0	PA_RAMP	Sets the PA ramp rate.																		
		The PA will ramp at the programmed rate until it reaches the level indicated by the PA_LEVEL_MCR (Address 0x307) setting. The ramp rate is dependent on the programmed data rate.																		
		<table><tr><th>PA_RAMP</th><th>Ramp rate</th></tr><tr><td>000</td><td>Reserved</td></tr><tr><td>001</td><td>256 codes per data bit</td></tr><tr><td>010</td><td>128 codes per data bit</td></tr><tr><td>011</td><td>64 codes per data bit</td></tr><tr><td>100</td><td>32 codes per data bit</td></tr><tr><td>101</td><td>16 codes per data bit</td></tr><tr><td>110</td><td>8 codes per data bit</td></tr><tr><td>111</td><td>4 codes per data bit</td></tr></table>	PA_RAMP	Ramp rate	000	Reserved	001	256 codes per data bit	010	128 codes per data bit	011	64 codes per data bit	100	32 codes per data bit	101	16 codes per data bit	110	8 codes per data bit	111	4 codes per data bit
		PA_RAMP	Ramp rate																	
		000	Reserved																	
		001	256 codes per data bit																	
		010	128 codes per data bit																	
		011	64 codes per data bit																	
		100	32 codes per data bit																	
		101	16 codes per data bit																	
		110	8 codes per data bit																	
		111	4 codes per data bit																	

Table 143: 0x115 RADIO\_CFG\_9

Bits	Name	Description												
7 to 6	IFBW	Sets the receiver IF filter bandwidth. <table><tr><th>IFBW</th><th>IF Bandwidth</th></tr><tr><td>00</td><td>100KHz</td></tr><tr><td>01</td><td>150KHz</td></tr><tr><td>10</td><td>200KHz</td></tr><tr><td>11</td><td>300KHz</td></tr></table>	IFBW	IF Bandwidth	00	100KHz	01	150KHz	10	200KHz	11	300KHz		
		IFBW	IF Bandwidth											
		00	100KHz											
		01	150KHz											
		10	200KHz											
		11	300KHz											
Note that setting an IF filter bandwidth of 300kHz automatically changes the receiver IF frequency from 200kHz to 300kHz:														
5 to 3	MOD_SCHEME	Sets the transmitter modulation. <table><tr><th>MOD_SCHEME</th><th>Modulation scheme</th></tr><tr><td>00</td><td>2-Level 2FSK/MSK</td></tr><tr><td>01</td><td>2-Level GFSK/GSMK</td></tr><tr><td>10</td><td>OOK</td></tr><tr><td>11</td><td>Carrier Only</td></tr><tr><td>100 to 111</td><td>Reserved</td></tr></table>	MOD_SCHEME	Modulation scheme	00	2-Level 2FSK/MSK	01	2-Level GFSK/GSMK	10	OOK	11	Carrier Only	100 to 111	Reserved
		MOD_SCHEME	Modulation scheme											
		00	2-Level 2FSK/MSK											
		01	2-Level GFSK/GSMK											
		10	OOK											
		11	Carrier Only											
100 to 111	Reserved													
2 to 0	DEMOD_SCHEME	Sets the receiver demodulation. <table><tr><th>DEMOD_SCHEME</th><th>Demodulation Scheme</th></tr><tr><td>00</td><td>2FSK/GFSK/MSK/GMSK</td></tr><tr><td>01</td><td>Reserved</td></tr><tr><td>10</td><td>OOK</td></tr><tr><td>011 to 111</td><td>Reserved</td></tr></table>	DEMOD_SCHEME	Demodulation Scheme	00	2FSK/GFSK/MSK/GMSK	01	Reserved	10	OOK	011 to 111	Reserved		
		DEMOD_SCHEME	Demodulation Scheme											
		00	2FSK/GFSK/MSK/GMSK											
		01	Reserved											
		10	OOK											
		011 to 111	Reserved											

Table 144: 0x116 RADIO\_CFG\_10

Bits	Name	Description
7 to 5	Reserved	Set to 0
4	AFC_POLARITY	Sets the AFC polarity: 0: Invert AFC polarity 1: Default AFC polarity <b>afc_polarity should be set to 0</b>
3 to 2	AFC_SCHEME	00: Reserved 01: Reserved 10 : AFC Mode 1 11: Reserved <b>afc_scheme should be set to 2</b>
1 to 0	AFC_LOCK_MODE	Sets the AFC mode. 00: Free Running: AFC is free running 01: Disabled: AFC is disabled 10: Hold AFC: AFC is paused 11: Lock: AFC locks after preamble or sync word (only locks on sync word if PREAMBLE_MATCH =0)

Table 145: 0x117 RADIO\_CFG\_11

Bits	Name	Description
7 to 4	AFC_KP	Sets the AFC PI controller proportional gain in 2FSK/GFSK/MSK/GMSK ,the <b>recommended value is 0x3</b> . In OOK demodulation, this setting is used to control the OOK threshold loop and the <b>recommended value is 0x3</b> . 0000:2^0 0001:2^1 0010:2^2 : : 1111:2^15
3 to 0	AFC_KI	Sets the AFC PI controller integral gain in 2FSK/GFSK/MSK/GMSK and the <b>recommended value is 0x7</b> . In OOK modulation this setting is used to control the OOK threshold loop and the <b>recommended value is 0x6</b> . 0000:2^0 0001:2^1 0010:2^2 : : 1111:2^15

Table 146: 0x118 IMAGE\_REJECT\_CAL\_PHASE

Bits	Name	Description										
7	Reserved	Set to 0										
6 to 0	IMAGE_REJECT_CAL_PHASE	<div>Sets the I/Q Phase adjustment .</div> <table><tr><th>IMAGE_REJECT_CAL_PHASE</th><th>I/Q Phase adjustment</th></tr><tr><td>0000000</td><td>0</td></tr><tr><td>0000001</td><td>1</td></tr><tr><td>:</td><td>:</td></tr><tr><td>1111111</td><td>63</td></tr></table>	IMAGE_REJECT_CAL_PHASE	I/Q Phase adjustment	0000000	0	0000001	1	:	:	1111111	63
IMAGE_REJECT_CAL_PHASE	I/Q Phase adjustment											
0000000	0											
0000001	1											
:	:											
1111111	63											

Table 147: 0x119 IMAGE\_REJECT\_CAL\_AMPLITUDE

Bits	Name	Description										
7	Reserved											
6 to 0	IMAGE_REJECT_CAL_AMPLITUDE	<div>Sets the I/Q Amplitude adjustment</div> <table><tr><th>IMAGE_REJECT_CAL_AMPLITUDE</th><th>I/Q Amplitude adjustment</th></tr><tr><td>0000000</td><td>0</td></tr><tr><td>0000001</td><td>1</td></tr><tr><td>:</td><td>:</td></tr><tr><td>1111111</td><td>63</td></tr></table>	IMAGE_REJECT_CAL_AMPLITUDE	I/Q Amplitude adjustment	0000000	0	0000001	1	:	:	1111111	63
IMAGE_REJECT_CAL_AMPLITUDE	I/Q Amplitude adjustment											
0000000	0											
0000001	1											
:	:											
1111111	63											



**Table 148: 0x11A MODE\_CONTROL**

Bits	Name	Description
7	SWM_EN	1: Smart wake mode enabled 0: Smart wake mode disabled
6	BB_CAL	1 : IF Filter calibration enabled 0 : IF Filter calibration disabled  The IF filter calibrations is automatically performed on the transition from the PHY_OFF to the PHY_ON state if this bit is set.
5	SWM_RSSI_QUAL	1 : RSSI qualify in low power mode enabled 0 : RSSI qualify in low power mode disabled
4	TX_AUTO_TURNAROUND	If TX_TO_RX_AUTO_TURNAROUND = 1, the device automatically transitions to the PHY_RX state at the end of a packet transmission, on the same RF channel frequency. If TX_TO_RX_AUTO_TURNAROUND = 0, this operation is disabled. TX_TO_RX_AUTO_TURNAROUND is only available in packet mode.
3	RX_AUTO_TURNAROUND	If RX_TO_TX_AUTO_TURNAROUND = 1, the device automatically transitions to the PHY_TX state at the end of a valid packet reception, on the same RF channel frequency. If RX_TO_TX_AUTO_TURNAROUND = 0, this operation is disabled. RX_TO_TX_AUTO_TURNAROUND is only available in packet mode.
2	CUSTOM_TRX_SYNTH_LOCK_TIME_EN	1: Use the custom synthesizer lock time defined in register 0x13E and 0x13F 0: default synthesizer lock time
1	EXT_LNA_EN	1: External LNA enable signal on ATB4 is enabled. The signal is logic high while the UHF Transceiver is in the PHY_RX state and logic low while in any other non-sleep state 0: External LNA enable signal on ATB4 is disabled.
0	EXT_PA_EN	1: External PA enable signal on ATB3 is enabled. The signal is logic high while the UHF Transceiver is in the PHY_TX state and logic low while in any other non-sleep state. 0: External PA enable signal on ATB3 is disabled.

Table 149: 0x11B PREAMBLE\_MATCH

Bits	Name	Description	
7	EXT_PA_LNA_CONFIG	1: External PA signal on XOSC32KP_GP5_ATB1 and external LNA signal on XOSC32KN_ATB2 (V <sub>DD</sub> logic outputs) 0: External PA signal on ADCIN_ATB3 and external LNA signal on ATB4 (1.8 V logic outputs)	
6 to 4	Reserved	Set to 0	
3 to 0	PREAMBLE_MATCH		
		<b>PREAMBLE_MATCH</b>	<b>Description</b>
		12(1100)	0 errors allowed.
		11(1011)	One erroneous bit-pair allowed in 12 bit-pairs.
		10(1010)	Two erroneous bit-pairs allowed in 12 bit-pairs.
		9(1001)	Three erroneous bit-pairs allowed in 12 bit-pairs.
		8 (1000)	Four erroneous bit-pairs allowed in 12 bit-pairs.
		0 (00)	Preamble detection disabled.
		1 to 7 ( 01 to 111)	Not recommended.
13 to 15 (1101 to 1111)	Reserved.		

Table 150: 0x11C SYMBOL\_MODE

Bits	Name	Description	
7	Reserved		
6	MANCHESTER_ENC	1: Manchester encoding/decoding enabled 0: Manchester encoding/decoding disabled	
5	PROG_CRC_EN	1: Programmable CRC enabled 0: Programmable CRC disabled	
4	EIGHT_TEN_ENC	1: 8b/10b encoding/decoding enabled 0: 8b/10b encoding/decoding disabled	
3	DATA_WHITENING	1: Data whitening/de-whitening enabled 0: Data whitening/de-whitening disabled	
2 to 0	SYMBOL_LENGTH		
		<b>SYMBOL_LENGTH</b>	
		000	8 bit (recommended except when 8b/10b is being used)
		001	10 bit (for 8b/10b encoding)
		010 to 111	Reserved

**Table 151: 0x11D PREAMBLE\_LEN**

Bits	Name	Description
7 to 0	PREAMBLE_LEN	Length of preamble in bytes. Example a value of decimal 3 results in a preamble of 24 bits.

**Table 152: 0x11E CRC\_POLY\_0**

Bits	Name	Description
7 to 0	CRC_POLY [7:0]	Lower byte of CRC_POLY[15:0], which sets the CRC polynomial.

**Table 153: 0x11F CRC\_POLY\_1**

Bits	Name	Description
7 to 0	CRC_POLY_1[15:8]	Upper byte of CRC_POLY[15:0] which sets the CRC polynomial.

**Table 154: 0x120 SYNC CONTROL**

Bits	Name	Description														
7 to 6	SYNC_ERROR_TOL	Sets the sync word error tolerance in bits.														
		<table><tr><th>SYNC_ERROR_TOL</th><th>Bit error tolerance</th></tr><tr><td>00</td><td>0 bit errors allowed</td></tr><tr><td>01</td><td>1 bit error allowed</td></tr><tr><td>10</td><td>2 bit error allowed</td></tr><tr><td>11</td><td>3 bit error allowed</td></tr></table>	SYNC_ERROR_TOL	Bit error tolerance	00	0 bit errors allowed	01	1 bit error allowed	10	2 bit error allowed	11	3 bit error allowed				
		SYNC_ERROR_TOL	Bit error tolerance													
		00	0 bit errors allowed													
		01	1 bit error allowed													
		10	2 bit error allowed													
11	3 bit error allowed															
5	Reserved	Reserved ,Set to 0														
4 to 0	SYNC_WORD_LENGTH	Sets the sync word length in bits. 24 bits is the maximum. The sync word matching length can be any value up to 24 bits, but the transmitted sync word pattern is a multiple of 8 bits. Hence, for non-byte-length sync words, the transmitted sync pattern should be filled out with the preamble pattern.														
		<table><tr><td>00000</td><td>No Sync Word detected</td></tr><tr><td>00001</td><td>1 bit syncword</td></tr><tr><td>00010</td><td>2 bit syncword</td></tr><tr><td>:</td><td>:</td></tr><tr><td>:</td><td>:</td></tr><tr><td>10111</td><td>23 bit syncword</td></tr><tr><td>11000</td><td>24 bit syncword</td></tr></table>	00000	No Sync Word detected	00001	1 bit syncword	00010	2 bit syncword	:	:	:	:	10111	23 bit syncword	11000	24 bit syncword
		00000	No Sync Word detected													
		00001	1 bit syncword													
		00010	2 bit syncword													
		:	:													
		:	:													
		10111	23 bit syncword													
11000	24 bit syncword															

**Table 155: 0x121 SYNC\_BYTE\_0**

Bits	Name	Description
7 to 0	SYNC_BYTE[7:0]	<p>Lower byte of sync word pattern. The sync word pattern is transmitted most significant bit first starting with SYNC_BYTE_0.</p> <p>For non-byte length sync words the remainder of the least significant byte should be stuffed with preamble.</p> <p>If SYNC_WORD_LENGTH length is &gt;16 bits then SYNC_BYTE_0, SYNC_BYTE_1 AND SYNC_BYTE_2 are all transmitted for a total of 24 bits.</p> <p>If SYNC_WORD_LENGTH is between 8 and 15 then SYNC_BYTE_1 AND SYNC_BYTE_2 are transmitted.</p> <p>If SYNC_WORD_LENGTH is between 1 and 7 then SYNC_BYTE_2 is transmitted for a total of 8 bits.</p> <p>If the SYNC_WORD_LENGTH is 0 then no sync bytes are transmitted.</p>

**Table 156: 0x122 SYNC\_BYTE\_1**

Bits	Name	Description
7 to 0	SYNC_BYTE[15:8]	<p>Mid byte of sync word pattern.</p> <p>See Table 155: 0x121 SYNC_BYTE_0 for further details</p>

**Table 157: 0x123 SYNC\_BYTE\_2**

Bits	Name	Description
7 to 0	SYNC_BYTE[23:16]	<p>Upper byte of sync word pattern.</p> <p>See Table 155: 0x121 SYNC_BYTE_0 for further details</p>

**Table 158: 0x124 TX\_BASE\_ADR**

Bits	Name	Description
7 to 0	TX_BASE_ADR	Address in Packet RAM of transmit packet. This address indicates to the communications processor the location of the first byte of the transmit packet

**Table 159: 0x125 RX\_BASE\_ADR**

Bits	Name	Description
7 to 0	RX_BASE_ADR	Address in Packet RAM of receive packet. The communications processor will write any qualified received packet to Packet RAM, starting at this memory location.

Table 160: 0x126 PACKET\_LENGTH\_CONTROL

Bits	Name	Description																		
7	DATA_BYTE	Over the air arrangement of each transmitted Packet RAM byte. Byte transmitted either MSB or LSB first. The same setting should be used on the Tx and Rx side of the link. 1: Data byte MSB first 0: Data byte LSB first																		
6	PACKET_LEN	1: Fixed packet length mode. Fixed packet length in Tx and Rx, given by PACKET_LENGTH_MAX. 0: Variable packet length mode. In Rx mode packet length is given by first byte in Packet RAM. In Tx mode the packet length is given by PACKET_LENGTH_MAX.																		
5	CRC_EN	1: Append CRC in transmit mode. Check CRC in receive mode. 0: No CRC addition in transmit mode. No CRC check in receive mode.																		
4 to 3	DATA MODE	Sets the packet mode or SPORT mode for transmit and receive data. <table><tr><th>DATA_MODE</th><th>Description</th></tr><tr><td>0</td><td>Packet Mode enabled</td></tr><tr><td>1</td><td>SPORT mode enabled. GP4 interrupt enabled on preamble detection. Rx Data enabled on preamble detection.</td></tr><tr><td>2</td><td>SPORT mode enabled. GP4 interrupt enabled on sync word detection. Rx Data enabled on preamble detection.</td></tr><tr><td>3</td><td>Reserved</td></tr></table>	DATA_MODE	Description	0	Packet Mode enabled	1	SPORT mode enabled. GP4 interrupt enabled on preamble detection. Rx Data enabled on preamble detection.	2	SPORT mode enabled. GP4 interrupt enabled on sync word detection. Rx Data enabled on preamble detection.	3	Reserved								
DATA_MODE	Description																			
0	Packet Mode enabled																			
1	SPORT mode enabled. GP4 interrupt enabled on preamble detection. Rx Data enabled on preamble detection.																			
2	SPORT mode enabled. GP4 interrupt enabled on sync word detection. Rx Data enabled on preamble detection.																			
3	Reserved																			
2 to 0	LENGTH_OFFSET	Offset value in bytes that is added to the packet length field value so the communications processor knows the correct number of bytes to read. (does not include CRC). The communications processor calculates the actual received payload length as Rx Payload Length = Length + LENGTH_OFFSET -4 Where Length is the length field ( the first byte in the received payload) <table><tr><th>LENGTH_OFFSET</th><th>Bytes</th></tr><tr><td>000</td><td>-4</td></tr><tr><td>001</td><td>-3</td></tr><tr><td>010</td><td>-2</td></tr><tr><td>011</td><td>-1</td></tr><tr><td>100</td><td>0</td></tr><tr><td>101</td><td>1</td></tr><tr><td>110</td><td>2</td></tr><tr><td>111</td><td>3</td></tr></table>	LENGTH_OFFSET	Bytes	000	-4	001	-3	010	-2	011	-1	100	0	101	1	110	2	111	3
LENGTH_OFFSET	Bytes																			
000	-4																			
001	-3																			
010	-2																			
011	-1																			
100	0																			
101	1																			
110	2																			
111	3																			

**Table 161: 0x127 PACKET\_LENGTH\_MAX**

Bits	Name	Description
7 to 0	PACKET_LENGTH_MAX	<p>If variable packet length mode is used (PACKET_LENGTH_CONTROL = 0), then PACKET_LENGTH_MAX sets the maximum packet length in bytes.</p> <p>If fixed packet length mode is used (PACKET_LENGTH_CONTROL = 1), then PACKET_LENGTH_MAX sets the length of the fixed transmit and receive packet in bytes.</p> <p>Note that the packet length is defined as the number of bytes from the end of the sync word to the start of the CRC. It also does not include the LENGTH_OFFSET value.</p>

**Table 162: 0x128 STATIC\_REG\_FIX**

Bits	Name	Description																					
7 to 0	STATIC_REG_FIX	<p>The UHF transceiver has the ability to implement automatic static register fixes from BBRAM memory to MCR memory. This feature allows a maximum of nine MCR registers to be programmed via BBRAM memory. This feature is useful if MCR registers must be configured for optimum receiver performance in low power mode.</p> <p>The STATIC_REG_FIX value is an address pointer to any BBRAM memory address between 0x12A and 0x13D. For example, to point to BBRAM Address 0x12B, set STATIC_REG_FIX= 0x2B.</p> <p>If STATIC_REG_FIX = 0x00, then static register fixes are disabled.</p> <p>If STATIC_REG_FIX is nonzero, the communications processor looks for the MCR address and corresponding data at the BBRAM address beginning at STATIC_REG_FIX.</p> <p>Example: write 0x46 to MCR Register 0x35E and write 0x78 to MCR Register 0x35F. Set STATIC_REG_FIX = 0x2B.</p> <table border="1"> <thead> <tr> <th>BBRAM Register</th><th>Data</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0x128 (STATIC_REG_FIX)</td><td>0x2B</td><td>Pointer to BBRAM Address 0x12B</td></tr> <tr> <td>0x12B</td><td>0x5E</td><td>MCR Address 1</td></tr> <tr> <td>0x12C</td><td>0x46</td><td>Data to write to MCR Address 1</td></tr> <tr> <td>0x12D</td><td>0x5F</td><td>MCR Address 2</td></tr> <tr> <td>0x12E</td><td>0x78</td><td>Data to write to MCR Address 2</td></tr> <tr> <td>0x12F</td><td>0x00</td><td>Ends static MCR register fixes</td></tr> </tbody> </table>	BBRAM Register	Data	Description	0x128 (STATIC_REG_FIX)	0x2B	Pointer to BBRAM Address 0x12B	0x12B	0x5E	MCR Address 1	0x12C	0x46	Data to write to MCR Address 1	0x12D	0x5F	MCR Address 2	0x12E	0x78	Data to write to MCR Address 2	0x12F	0x00	Ends static MCR register fixes
BBRAM Register	Data	Description																					
0x128 (STATIC_REG_FIX)	0x2B	Pointer to BBRAM Address 0x12B																					
0x12B	0x5E	MCR Address 1																					
0x12C	0x46	Data to write to MCR Address 1																					
0x12D	0x5F	MCR Address 2																					
0x12E	0x78	Data to write to MCR Address 2																					
0x12F	0x00	Ends static MCR register fixes																					

**Table 163: 0x129 ADDRESS\_MATCH\_OFFSET**

Bits	Name	Description
7 to 0	ADDRESS_MATCH_OFFSET	Location of first byte of address information in Packet RAM

**Table 164: 0x12A ADDRESS\_LENGTH**

Bits	Name	Description
7 to 0	ADDRESS_LENGTH	Number of bytes in each address field ( $N_{ADR\_1}$ ) Set to 0 if address filtering is not being used.

**Table 165: 0x12B to 0x13D ADDRESS FILTERING**

Address	Bits	Description
0x12B	[7:0]	Address 1 match byte 0
0x12C	[7:0]	Address 1 mask byte 0
0x12D	[7:0]	Address 1 match byte 1
0x12E	[7:0]	Address 1 mask byte 1
:		:
	[7:0]	Address Match byte NADR-1
	[7:0]	Address Mask byte NADR-1
	[7:0]	0x00 to end or NADR for another address check sequence.

**Table 166: 0x13E RX\_SYNTH\_LOCK\_TIME**

Bits	Name	Description
4 to 0	RX_SYNTH_LOCK_TIME	Allows the use of a custom synthesizer lock time counter in receive mode in conjunction with the CUSTOM_TRX_SYNTH_LOCK_TIME_EN setting in the MODE_CONTROL register. Applies after VCO calibration is complete. Each bit equates to a 2 $\mu$ s increment.

**Table 167: 0x13F TX\_SYNTH\_LOCK\_TIME**

Bits	Name	Description
4 to 0	TX_SYNTH_LOCK_TIME	Allows the use of a custom synthesizer lock time counter in transmit mode in conjunction with the CUSTOM_TRX_SYNTH_LOCK_TIME_EN setting in the MODE_CONTROL register. Applies after VCO calibration is complete. Each bit equates to a 2 $\mu$ s increment.

## MCR Register Description

Note that the MCR register settings are not retained when the device enters PHY\_SLEEP.

**Table 168: 0x307 PA\_LEVEL\_MCR**

Bits	Name	Description
7 to 6	Reserved	
5 to 0	PA_LEVEL_MCR	Power Amplifier Level. If PA Ramp is enabled the PA will ramp to this target level. The PA level can be set in the range 0 to 63. The PA level (with less resolution) can also be set via the BBRAM so the MCR setting should only be used if more resolution is required.

**Table 169: 0x30C WUC\_CONFIG\_HIGH**

Bits	Name	Description																											
7	Reserved	Reserved																											
6	WUC_BGAP	Set to 0																											
5	WUC_SYNTH	Set to 0																											
4	WUC_LDO_DIG	Set to 0																											
3	WUC_XTO26M	Set to 0																											
2 to 0	WUC_PRESCALER	<table> <tr> <th>WUC_PRESCALER</th><th>32.768KHz Divider</th><th>Tick Period</th></tr> <tr> <td>000</td><td>1</td><td>30.52 <math>\mu</math>s</td></tr> <tr> <td>001</td><td>4</td><td>122.1 <math>\mu</math>s</td></tr> <tr> <td>010</td><td>8</td><td>244.1 <math>\mu</math>s</td></tr> <tr> <td>011</td><td>16</td><td>488.3 <math>\mu</math>s</td></tr> <tr> <td>100</td><td>128</td><td>3.91 s</td></tr> <tr> <td>101</td><td>1024</td><td>31.25 ms</td></tr> <tr> <td>110</td><td>8192</td><td>250 ms</td></tr> <tr> <td>111</td><td>65536</td><td>2000 ms</td></tr> </table>	WUC_PRESCALER	32.768KHz Divider	Tick Period	000	1	30.52 $\mu$ s	001	4	122.1 $\mu$ s	010	8	244.1 $\mu$ s	011	16	488.3 $\mu$ s	100	128	3.91 s	101	1024	31.25 ms	110	8192	250 ms	111	65536	2000 ms
WUC_PRESCALER	32.768KHz Divider	Tick Period																											
000	1	30.52 $\mu$ s																											
001	4	122.1 $\mu$ s																											
010	8	244.1 $\mu$ s																											
011	16	488.3 $\mu$ s																											
100	128	3.91 s																											
101	1024	31.25 ms																											
110	8192	250 ms																											
111	65536	2000 ms																											

Note: WUC\_CONFIG\_LOW should never be written to without updating WUC\_CONFIG\_HIGH first.



**Table 170: 0x30D WUC\_CONFIG\_LOW**

Bits	Name	Description
7	Reserved	Reserved
6	WUC_RCOSC_EN	1: Enable RCOSC32K 0: Disable RCOSC32K
5	WUC_XOSC32K_EN	1: Enable XOSC32K 0: Disable XOSC32K
4	WUC_CLKSEL	Select WUC timer clock source: 1: RC 32.768kHz oscillator 0: External crystal oscillator
3	WUC_BBRAM_EN	1: Enable power to BBRAM during PHY_SLEEP 0: Disable power to BBRAM during PHY_SLEEP
2 to 1	Reserved	Reserved
0	WUC_WUC_ARM	1: Enable wake-up on WUC time-out event 0: Disable wake-up on WUC time-out event

Note: WUC\_CONFIG\_LOW should never be written to without updating WUC\_CONFIG\_HIGH first.

**Table 171: 0x30E WUC\_VALUE\_HIGH**

Bits	Name	Description
7	WUC_TIMER_VALUE[15:8]	WUC timer reload value, Bits[15:8] of [15:0]. A wake-up event is triggered when the WUC unit has been enabled and the timer has counted down to 0. The timer is clocked with the prescaler output rate. An update to this register becomes effective only after wake_timer_value_low is written.

Note: Updates to WUC\_VALUE\_HIGH become effective only after WUC\_VALUE\_LOW has been written.

**Table 172: 0x30F WUC\_VALUE\_LOW**

Bits	Name	Description
7 to 0	WUC_TIMER_VALUE[7:0]	WUC timer reload value, Bits[7:0] of [15:0]. A wake-up event is triggered when the WUC unit has been enabled and the timer has counted down to 0. The timer is clocked with the prescaler output rate.

Note: WUC\_VALUE\_LOW should never be written to without updating WUC\_VALUE\_HIGH first.

Table 173: 0x310 WUC\_FLAG\_RESET

Bits	Name	Description
7 to 2	Reserved	
1	WUC_RCOSC_CAL_EN	1: Enable RCOSC32K calibration 0: Disable RCOSC32K calibration
0	WUC_FLAG_RESET	1: Reset bits WUC_TMR_PRIM_TOFLAG and WUC_PORFLAG(Address 0x311) 0: Normal operation

Table 174: 0x311 WUC\_STATUS

Bits	Name	Description
7	Reserved	Reserved
6	WUC_RCOSC_CAL_ERROR	1: RCOSC32K calibration exited with error 0: without error (only valid if WUC_RCOSC_CAL_EN = 1)
5	WUC_RCOSC_CAL_READY	1: RCOSC32K calibration finished 0: in progress (only valid if WUC_RCOSC_CAL_EN = 1)
4	XOSC32K_RDY	1: XOSC32K oscillator has settled 0: not settled (only valid if WUC_CONFIG_LOW_XOSC32K_EN = 1)
3	XOSC32K_OUT	output signal of XOSC32K oscillator (instantaneous)
2	WUC_PORFLAG	1: Chip cold start event has been registered 0: not registered
1	WUC_TMR_PRIM_TOFLAG	1: WUC time-out event has been registered 0: not registered (The o/p of a latch triggered by a time out event)
0	WUC_TMR_PRIM_TOEVENT	1: WUC time-out event is present 0: not present (This bit is set when the counter has reached zero. It is not latched.)

Table 175: 0x312 RSSI\_READBACK

Bits	Name	Description
7 to 0	RSSI_READBACK	Receive input power. After reception of a packet the RSSI_READBACK value is valid.  RSSI (dBm) = RSSI_READBACK -107

Table 176: 0x315 MAX\_AFC\_RANGE

Bits	Name	Description
7 to 0	MAX_AFC_RANGE	Limits the AFC pull-in Range. Automatically set by the communications processor on transitioning into the PHY_RX state. Should be set to less than half the receive IF filter bandwidth. Example: IF bandwidth = 200kHz, AFC pull-in range = +/-100kHz (MAX_AFC_RANGE =100)

**Table 177: 0x319 IMAGE\_REJECT\_CAL\_CONFIG**

Bits	Name	Description
7 to 6	Reserved	
5	IMAGE_REJECT_CAL_OVWRT_EN	Overwrite control for Image Reject Calibration results
4 to 3	IMAGE_REJECT_FREQUENCY	Set the fundamental frequency of the IR Cal Source. A harmonic of this frequency can be used as an internal RF signal source for the image rejection calibration. 00 :IR Cal source disabled in XTAL divider 01 :IR calibration source fundamental frequency = XTAL/4 10 : IR calibration source fundamental frequency = XTAL/8 11 : IR calibration source fundamental frequency = XTAL/16
2 to 0	IMAGE_REJECT_POWER	Set power level of IR Cal source 000: IR calibration source disabled at mixer input 001: power level = min 010: power level = min 011: power level = min × 2 100: power level = min × 2 101: power level = min × 3 110: power level = min × 3 111: power level = min × 4

**Table 178: 0x322 CHIP\_SHUTDOWN**

Bits	Name	Description
7 to 1	Reserved	
0	CHIP_SHTDN_REQ	WUC chip state control flag 1: Invoke chip shutdown. Note that CS must also be high to initiate a shutdown. 0: remain in active state

**Table 179: 0x324 POWERDOWN\_RX**

Bits	Name	Description
7 to 5	Reserved	
4	ADC_PD_N	1: LNA enabled 0: LNA disabled
3	RSSI_PD_N	1: RSSI enabled 0: RSSI disabled
2	RXBBFILT_PD_N	1: IF filter enabled 0: IF filter disabled
1	RXMIXER_PD_N	1: Mixer enabled 0: Mixer disabled
0	LNA_PD_N	1: LNA enabled 0: LNA disabled

**Table 180: 0x325 POWERDOWN\_AUX**

Bits	Name	Description
7 to 2	Reserved	Reserved
1	TEMPMON_PD_N	1: Enable temperature monitor 0: Disable temperature monitor
0	BATTMON_PD_N	1: Enable battery monitor 0: Disable battery monitor

**Table 181: 0x327 ADC\_READBACK\_HIGH**

Bits	Name	Description
7 to 6	Reserved	Reads zero
5 to 0	ADC_READBACK[7:2]	ADC readback MSBs

**Table 182: 0x328 ADC\_READBACK\_LOW**

Bits	Name	Description
7 to 6	ADC_READBACK[1:0]	ADC readback LSBs
5 to 0	Reserved	Reserved

**Table 183: 0x32D BATTERY\_MONITOR\_THRESHOLD\_VOLTAGE**

Bits	Name	Description
7 to 5	Reserved	
4 to 0	BATTMON_VOLTAGE	Battery Monitor threshold voltage sets the alarm level for the battery monitor. The alarm is raised by interrupt. Battery Monitor trip voltage, $V_{trip} = 1.7V + 62mV \times battmon\_voltage$

**Table 184: 0x32E EXT\_UC\_CLK\_DIVIDE**

Bits	Name	Description
7 to 4	Reserved	
3 to 0	EXT_UC_CLK_DIVIDE	Optional output clock frequency on GP5. Output Frequency = XTAL/EXT_UC_CLK_DIVIDE To disable set EXT_UC_CLK_DIVIDE = 0.

**Table 185: 0x32F AGC\_CLOCK\_DIVIDE**

Bits	Name	Description
7 to 0	AGC_CLOCK_DIVIDE	AGC Clk Divider for 2FSK/GFSK/MSK/GMSK mode. The AGC rate is (26MHz/(16 x AGC_CLK_DIVIDE))

**Table 186: 0x336 INTERRUPT\_SOURCE\_0**

Bits	Name	Description
7	INTERRUPT_NUM_WAKEUPS	Asserts when the number of WUC wakeups (NUMBER_OF_WAKEUPS[15:0]) has reached the threshold (NUMBER_OF_WAKEUPS_IRQ_THRESHOLD[15:0])
6	INTERRUPT_SWM_RSSI_DET	Asserted when the measured RSSI during smart wake mode has exceeded the RSSI threshold value (SWM_RSSI_THRESH, Address 0x108)
5	INTERRUPT_AES_DONE	Asserted when an AES encryption or decryption command is complete; available only when the AES firmware module has been loaded to the UHF Transceiver program RAM
4	INTERRUPT_TX_EOF	Asserted when a packet has finished transmitting (packet mode only)
3	INTERRUPT_ADDRESS_MATCH	Asserted when a received packet has a valid address match (packet mode only)
2	INTERRUPT_CRC_CORRECT	Asserted when a received packet has the correct CRC (packet mode only)
1	INTERRUPT_SYNC_DETECT	Asserted when a qualified sync word has been detected in the received packet
0	INTERRUPT_PREMABLE_DETECT	Asserted when a qualified preamble has been detected in the received packet

**Table 187: 0x337 INTERRUPT\_SOURCE\_1**

Bits	Name	Description
7	BATTERY_ALARM	Battery voltage has dropped below user set threshold value
6	CMD_READY	Communications processor ready to accept a new command.
5	Reserved	Reserved
4	WUC_TIMEOUT	Wake up Timer has timed out
3 to 2	Reserved	Reserved
1	SPI_READY	SPI ready for access
0	CMD_FINISHED	Command has finished

**Table 188: 0x338 CALIBRATION\_CONTROL**

Bits	Name	Description
7 to 2	Reserved	
1	SYNTH_CAL_EN	1: Enables the synthesizer calibration state machine 0: Disables the synthesizer calibration state machine.
0	RXBB_CAL_EN	1: Enable receiver baseband filter (RXBB) calibration 0: Disable receiver baseband filter (RXBB) calibration

**Table 189: 0x339 CALIBRATION\_STATUS**

Bits	Name	Description
7 to 3	Reserved	
2	PA_RAMP_FINISHED	
1	SYNTH_CAL_READY	1: Synthesizer calibration finished successfully. 0: Synthesizer calibration in progress
0	RXBB_CAL_READY	Receive IF Filter calibration 1: Complete 0: in progress (valid while RXBB_CAL_EN = 1)

**Table 190: 0x345 RXBB\_CAL\_CALWRD\_READBACK**

Bits	Name	Description
5 to 0	RXBB_CAL_CALWRD	RXBB reference oscillator calibration word. Valid after RXBB calibration cycle has been completed.

**Table 191: 0x346 RXBB\_CAL\_CALWRD\_OVERWRITE**

Bits	Name	Description
6 to 1	RXBB_CAL_DCALWRD_OVWRT_IN	RXBB reference oscillator calibration overwrite word.
0	RXBB_CAL_DCALWRD_OVWRT_EN	1: Enable RXBB reference oscillator calibration word overwrite mode 0: Disable RXBB reference oscillator calibration word overwrite mode

**Table 192: 0x359 ADC\_CONFIG\_LOW**

Bits	Name	Description
7 to 4	Reserved	Reserved
3 to 2	ADC_REF_CHSEL	00: RSSI (default) 01: external AIN 10: temperature sensor 11: unused
1 to 0	ADC_REFERENCE_CONTROL	The following reference values are valid for a 3 V supply: 00: 1.85 V (default) 01: 1.95 V 10: 1.75 V 11: 1.65 V

**Table 193: 0x35A ADC\_CONFIG\_HIGH**

Bits	Name	Description
7	Reserved	
6 to 5	FILTERED_ADC_MODE	Filtering Mode 00: normal operation (no filter) 01: unfiltered AGC loop, filtered readback (update on MCR read) 10: unfiltered AGC loop, filtered readback (update at AGC clock rate) 11: filtered AGC loop, filtered readback
4	ADC_EXT_REF_ENB	Bring low to power down ADC Reference.
3 to 0	Reserved	Reserved

Table 194: 0x35B AGC\_OOK\_CONTROL

Bits	Name	Description
7 to 6	Reserved	
5 to 3	OOK_AGC_CLK_TRK	AGC update rate during tracking phase: $\text{AGC update rate} = \text{Fman} / [2^{(\text{ook\_agc\_iad\_trk}+1)}]$ Where: Fman = The Manchester symbol Rate. Manchester encoding is recommended for OOK Note: OOK_AGC_CLK_TRK must be $\geq$ OOK_AGC_CLK_ACQ
2 to 0	OOK_AGC_CLK_ACQ	AGC update rate during acquisition phase: $\text{AGC update rate} = \text{Fman} / [2^{(\text{ook\_agc\_clk\_acq}+1)}]$ Where: Fman = The Manchester symbol Rate. Manchester encoding is recommended for OOK. Note: OOK_AGC_CLK_TRK must be $\geq$ OOK_AGC_CLK_ACQ

Table 195: 0x35C AGC\_CONFIG

Bits	Name	Description
7 to 6	LNA_GAIN_CHANGE_ORDER	LNA gain change order
5 to 4	MIXER_GAIN_CHANGE_ORDER	Mixer gain change order
3 to 2	FILTER_GAIN_CHANGE_ORDER	Filter gain change order
1	ALLOW_EXTRA_LO_LNA_GAIN	Allow extra Low LNA gain setting
0	DISALLOW_MAX_GAIN	Disallow max AGC gain setting

Table 196: 0x35D AGC\_MODE

Bits	Name	Description
7	Reserved	
6 to 5	AGC_OPERATION_MCR	00: Free running AGC 01: Manual AGC 10: Freeze AGC 11: Lock AGC after preamble
4 to 3	LNA_GAIN	00: Low 01: Medium 10: High 11: Reserved
2	MIXER_GAIN	0: Low 1: High
1 to 0	FILTER_GAIN	00: Low 01: Medium 10: High 11: Reserved



**Table 197: 0x35E AGC\_LOW\_THRESHOLD**

Bits	Name	Description
7 to 0	AGC_LOW_THRESHOLD	AGC low threshold

**Table 198: 0x35F AGC\_HIGH\_THRESHOLD**

Bits	Name	Description
7 to 0	AGC_HIGH_THRESHOLD	AGC high threshold

**Table 199: 0x360 AGC\_GAIN\_STATUS**

Bits	Name	Description
7 to 5	Reserved	
4 to 3	LNA_GAIN_READBACK	00: Low 01: Medium 10: High 11: Reserved
2	MIXER_GAIN_READBACK	0: Low 1: High
1 to 0	FILTER_GAIN_READBACK	00: Low 01: Medium 10: High 11: Reserved

**Table 200: 0x372 FREQUENCY\_ERROR\_READBACK**

Bits	Name	Description
7 to 0	FREQUENCY_ERROR_READBACK	Frequency Error between received signal frequency and receive channel frequency = FREQUENCY_ERROR_READBACK × 100Hz. The FREQUENCY_ERROR_READBACK value is in twos complement format.

**Table 201: 0x3CB VCO\_BAND\_OVRW\_VAL**

Bits	Name	Description
7 to 0	VCO_BAND_OVRW_VAL	Overwrite value for the VCO frequency band. Active when VCO_BAND_OVRW_EN = 1

**Table 202: 0x3CC VCO\_AMPL\_OVRW\_VAL**

Bits	Name	Description
7 to 0	VCO_AMPL_OVRW_VAL	Overwrite value for the VCO bias current DAC. Active when VCO_AMPL_OVRW_EN = 1

**Table 203: 0x3CD VCO\_OVRW\_EN**

Bits	Name	Description
7 to 6	Reserved	Reserved
5 to 2	VCO_Q_AMP_REF	VCO amplitude level control reference DAC during Q phase.
1	VCO_AMPL_OVRW_EN	1:Enable VCO bias current DAC overwrite 0:Disable VCO bias current DAC overwrite
0	VCO_BAND_OVRW_EN	1:Enable VCO frequency band overwrite 0:Disable VCO frequency band overwrite

**Table 204: 0x3D0 VCO\_CAL\_CFG**

Bits	Name	Description
7 to 4	Reserved	Reserved
3 to 0	VCO_CAL_CFG	VCO calibration state machine configuration. Set VCO_CAL_CFG = 0xF to bypass the VCO calibration on the PHY_TX and PHY_RX transitions. Set VCO_CAL_CFG = 0x1 to enable the VCO calibrations on these transitions.

**Table 205: 0x3D2 XOSC\_CONFIG**

Bits	Name	Description
7 to 6	Reserved	Set to 0
5 to 3	XOSC_CAP_DAC	26MHz crystal oscillator (XOSC26M ) tuning capacitor control word
2 to 0	Reserved	Set to 0

**Table 206: 0x3DA VCO\_BAND\_READBACK**

Bits	Name	Description
7 to 0	VCO_BAND_READBACK	Read-back of the VCO bias current DAC after calibration

**Table 207: 0x3DB VCO\_AMPL\_READBACK**

Bits	Name	Description
7 to 0	VCO_AMPL_READBACK	Read-back of the VCO bias current DAC after calibration

**Table 208: 0x3F8 ANALOG\_TEST\_BUS**

Bits	Name	Description
7 to 0	ANALOG_TEST_SIX	To enable analog RSSI on ATB3 set ANALOG_TEST_BUS = 0x64 in conjunction with setting RSSI_TSTMUX_SEL = 0x3.

**Table 209: 0x3F9 RSSI\_TSTMUX\_SEL**

Bits	Name	Description
7 to 2	Reserved	
1 to 0	RSSI_TSTMUX_SEL	To enable analog RSSI on ATB3 set RSSI_TSTMUX_SEL = 0x3 in conjunction with setting ANALOG_TEST_BUS = 0x64.

**Table 210: 0x3FA GPIO\_CONFIGURE**

Bits	Name	Description
7 to 0	GPIO_CONFIGURE	0x00: default 0x21: Slicer output on GP5 (that is, bypass CDR) 0x40: Limiter outputs on GP0(Q) and GP1(I) 0x41: Filtered Limiter outputs on GP0(Q) and GP1(I) and un-filtered limiter outputs on GP2(Q) and IRQ_GP3(I) 0x50: packet transmit data from communications processor on GP0 0x53: PA ramp finished on GP0 0xA0: SPORT mode 0 0xA1: SPORT mode 1 0xA2: SPORT mode 2 0xA3: SPORT mode 3 0xA4: SPORT mode 4 0xA5: SPORT mode 5 0xA6: SPORT mode 6 0xA7: SPORT mode 7 0xA8: Sport Mode 8 0xC9: Test DAC output on GP0 (Also must set TEST_DAC_GAIN)

**Table 211: 0x3FD TEST\_DAC\_GAIN**

Bits	Name	Description
7 to 4	Reserved	Reserved
3 to 0	TEST_DAC_GAIN	Set TEST_DAC_GAIN = 0 when using the test DAC

# Digital I/Os

## Digital I/Os Functionality

The ADuCRF101 features up to 29 general-purpose bidirectional input/output (GPIO) pins. All of the GPIO pins have multiple functions, configurable by user code.

The GPIOs are grouped into ports, a port containing 8 GPIOs.

Each GPIO can be configured as input, output or open circuit. They also have an internal pull-up programmable resistor. All I/O pins are functional over the full supply range (VBAT = 1.8V to 3.6V) and the logic input voltages are specified as percentages of the supply:

$$V_{INL} = 0.2 \times IOVDD \text{ max}$$

$$V_{INH} = 0.7 \times IOVDD \text{ min}$$

Absolute maximum input voltage is IOVDD+0.3V.

Typical leakage current of the GPIOs configured as input or open circuit is 50nA per GPIO. All GPIOs have a drive capability of 4 mA.

When the ADuCRF101 enters a power-saving mode, the GPIO pins retain their state. Note that a driving peripheral will not be able to drive the pin. E.g. if the UART is driving the pin on entry to hibernate, it will be isolated from the pin and power gated. Its state and control will be restored on wake up.

## Digital Port Multiplex

This block provides control over the GPIO functionality of specified pins, since some of the pins have a choice to work as GPIO or have other specific functions. The GPIOs are grouped into ports with each port containing 8 GPIOs. The following tables detail the configuration modes for each GPIO.

Table 212: GPIO Multiplex Table

GPIO	Configuration Modes			
	00	01	10	11
<b>GP0</b>				
P0.0	GPIO	SPI MISO (SPI1)	-	-
P0.1	GPIO	SPI SCLK (SPI1)	-	-
P0.2	GPIO	SPI MOSI (SPI1)	-	PWM0
P0.3	GPIO/IRQ1	SPI CS0 <sup>1</sup> (SPI1)	-	PWM1
P0.4	GPIO/IRQ0	SPI CS1 <sup>1</sup> (SPI1)	ECLK OUT	-
P0.5	GPIO	SPI CS2 <sup>1</sup> (SPI1)	EXT CLK IN	-
P0.6	GPIO/IRQ2	SPI CS3 <sup>1</sup> (SPI1)	UART RTS	-
P0.7	GPIO/IRQ3	SPI CS4 <sup>1</sup> (SPI1)	UART CTS	-
<b>GP1</b>				
P1.0	GPIO/IRQ4	UART RXD	-	PWM2
P1.1	PORB	GPIO	UART TXD	PWM3
P1.2	-	GPIO	-	PWM4
P1.3	-	GPIO	-	PWM5
P1.4	GPIO/IRQ5	I2C0SCL	PWM6	-
P1.5	GPIO/IRQ6	I2C0SDA	PWM7	-
P1.6	GPIO/IRQ7	ADC_CONV_ST	-	PWM SYNC
<b>GP2</b>				
P2.0	MISO_RF (SP0)	GPIO	-	-
P2.1	SCLK_RF (SP0)	GPIO	-	-
P2.2	MOSI_RF (SP0)	GPIO	-	-
P2.3	CS_RF (SP0)	GPIO	-	-
P2.4	IRQ8_RF	GPIO	-	-
P2.5	-	-	GPIO	-
P2.6	-	GPIO	-	-
P2.7	GPIO	-	-	-

<sup>1</sup> SPI1 in slave mode uses CS0 as its chip select. In master mode, SPI1 controls the 5 chip select signals (P0.3 to P0.7) at the same time. The correct GPIO must be configured by user code as CS line(s) in the GPIO interface.

GPIO Multiplex Table Continued

GPIO	Configuration Modes			
	00	01	10	11
<b>GP3</b>				
P3.0	GPIO	-	-	PWM TRIPn
P3.1	GPIO	-	-	-
P3.2	ADIO_8	GPIO	PWM SYNC	MISO_RF
P3.3	ADIO_9	GPIO	PWM TRIPn	SCLK_RF
P3.4	ADIO_10	GPIO	-	-
P3.5	ADIO_11	GPIO	-	MOSI_RF
P3.6	-	GPIO	-	-
P3.7	-	GPIO	-	-
<b>GP4</b>				
P4.0	ADIO_0	GPIO	PWM0	-
P4.1	ADIO_1	GPIO	PWM1	-
P4.2	ADIO_2	GPIO	PWM2	CS_RF
P4.3	ADIO_3	GPIO	PWM3	-
P4.4	ADIO_4	GPIO	PWM4	-
P4.5	ADIO_5	GPIO	PWM5	-
P4.6	ADIO_6	GPIO	PWM6	-
P4.7	ADIO_7	GPIO	PWM7	-

## Inter-die Connectivity

10 internal digital signals are dedicated to the UHF Transceiver interface.

**Table 213: Internal Digital Signals**

GPIO	Bottom die signal	UHF Transceiver corresponding pin name
P2.0	MISO_RF	MISO
P2.1	SCLK_RF	SCLK
P2.2	MOSI_RF	MOSI
P2.3	CS_RF	CS
P2.4	IRQ8_RF	IRQ_GP3
P2.5	GPIO	GP5
P2.6	GPIO	GP0
P2.7	GPIO	GP4
P3.0	----	----
P3.1	----	----
P3.2	----	----
P3.3	----	----
P3.4	----	----
P3.5	----	----
P3.6	GPIO	GP1
P3.7	GPIO	GP2

## Access to Internal Signals

There are 10 connections between the two die that can be seen on external GPIOs for debug purposes using test modes. The GPIOs are listed in Table 214. The register controlling the test modes is described in the memory map section of the GPIO.

**Table 214. Digital Test Mode Outputs**

Top Die signal	External GPIO	Test mode register control RFTST
GP0	P1.0	DIR[0]
GP1	P1.1	DIR[1]
GP2	P0.6	DIR[2]
GP3	P0.7	DIR[3]
GP4	P1.4	DIR[4]
GP5	P1.5	DIR[5]
MISO	P0.1	DIR[7]
SCLK	P0.0	DIR[8]
MOSI	P0.3	DIR[9]
CS	P0.2	DIR[10]

The analog UHF transceiver signals can also be connected to GPIOs for debug or evaluation purposes:

**Table 215. Analog Test Mode Outputs**

Top Die signal	External GPIO	Test mode register control – RFTST
ATB1	P1.5	AN0
ATB2	P1.6	AN1
ATB3	P0.5	AN1
ATB4	P0.4	AN1



## Digital I/O Block Diagram

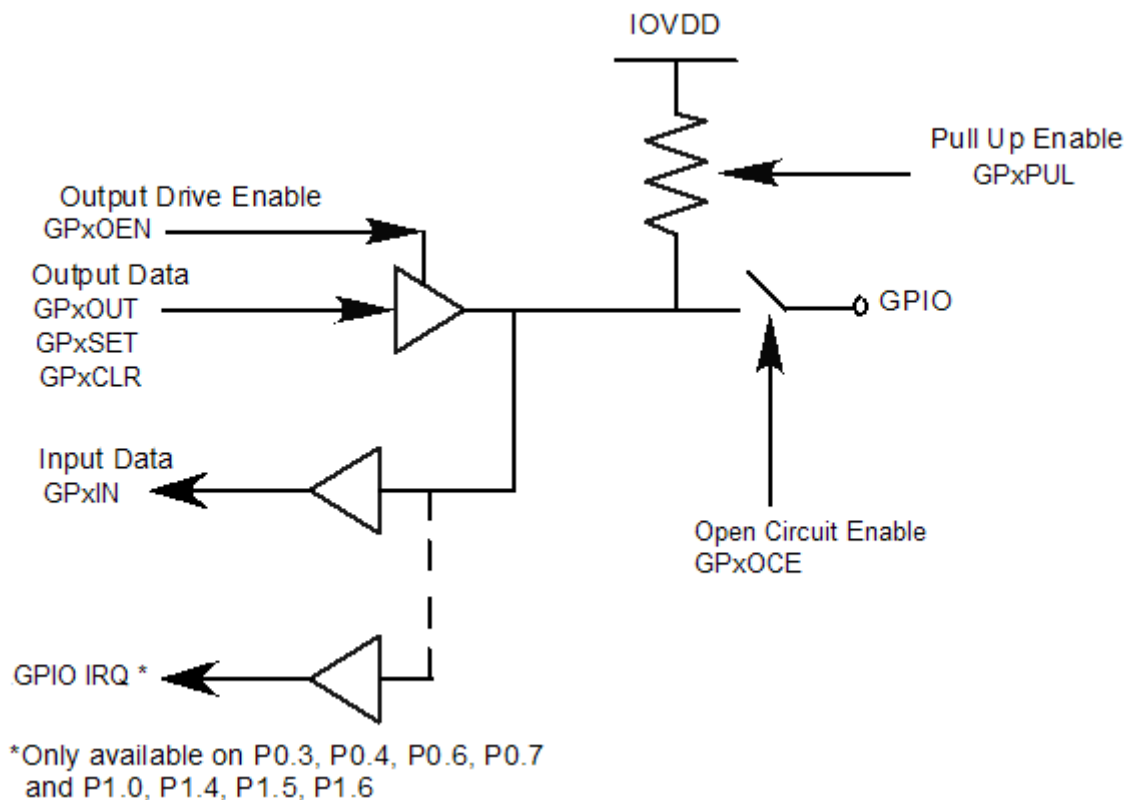


Figure 47: GPIO Structure

## Digital I/O Operation

### Digital Port Multiplexed Configuration:

The pin functions are configured using the GP0CON and GP1CON registers. GP0CON configures Port 0, P0.0 to P0.7 and GP1CON configures Port 1, P1.0 to P1.7.

External interrupts and input level signals (GPxIN) are available in any configuration modes except when the GPIOs are open circuit or output enabled (GPxOE = 1 and GPxOCE = 1).

Note that user code should not have to configure Port 2 and 3. These two ports are configured by default to interface with ADF7023 die. GP2CON and GP3CON are available for Port 2 and Port 3 configuration.

### IO Pull Up Enable

All GPIO pins have an internal pull-up resistor. Using the GPxPUL register it is possible to enable/disable pull-up registers on the pins when they are configured as inputs. The pull ups will be automatically disabled when the pad is set as an output or open circuit is enabled.

**IO Data In**

When configured as an input (by default), the GPIO input levels are available in GPxIN.

**Open Circuit Enable**

This disables the input paths if the pin is set as output. To disable the input and not drive the pin then the open circuit should be set and drive logic 1. External interrupts interrupt are not available when open circuit is enabled.

**IO Data Out**

When the GPIOs are configured as outputs, the values in GPxOUT are reflected on the GPIOs.

**Bit Set**

Bit Set mode is used to set one or more GPIO Data Out without affecting others within a port. Only the GPIO corresponding with write data bit equal to one will be set, the remaining GPIO will be unaffected.

**Bit Clear**

Bit Clear mode is used to clear one or more GPIO Data Out without affecting others within a port. Only the GPIO corresponding with write data bit equal to one will be cleared, the remaining GPIO will be unaffected.

**Bit Toggle**

Bit Toggle is used to toggle one or more GPIO Data Out without affecting others within a port. Only the GPIO corresponding to write data bit equal to one will be toggled, the remaining will be unaffected.

**IO Data Output Enable**

The Data Output path is enabled; the values in GPxOUT are reflected on the GPIOs.

# GPIO Memory Mapped Registers

**Table 216: GPIO Interface Memory Address Table**

Base Address: 0x40006000

Offset	Name	Description	Access	Default
0x0000	GP0CON	GPIO Port 0 configuration	RW	0x0000
0x0004	GP0OEN	GPIO port 0 output enable	RW	0x00
0x0008	GP0PUL	GPIO port 0 output pull up enable	RW	0xFF
0x000C	GP0OCE	GPIO Port 0 open circuit enable	RW	0x00
0x0014	GP0IN <sup>1</sup>	GPIO port 0 data input	R	xx
0x0018	GP0OUT	GPIO port 0 data out	RW	0x00
0x001C	GP0SET	GPIO port 0 data out set	W	0x00
0x0020	GP0CLR	GPIO port 0 data out clear	W	0x00
0x0024	GP0 TGL	GPIO port 0 data out toggle	W	0x00
0x0030	GP1CON	GPIO Port 1 configuration	RW	0x0000
0x0034	GP1OEN	GPIO port 1 output enable	RW	0x00
0x0038	GP1PUL	GPIO port 1 output pull up enable	RW	0x7F
0x003C	GP1OCE	GPIO Port 1 open circuit enable	RW	0x00
0x0044	GP1IN <sup>1</sup>	GPIO port 1 data input	R	xx
0x0048	GP1OUT	GPIO port 1 data out	RW	0x00
0x004C	GP1SET	GPIO port 1 data out set	W	0x00
0x0050	GP1CLR	GPIO port 1 data out clear	W	0x00
0x0054	GP1TGL	GPIO port 1 pin toggle	W	0x00
0x0060	GP2CON	GPIO Port 2 configuration	RW	0x0000
0x0064	GP2OEN	GPIO port 2 output enable	RW	0x00
0x0068	GP2PUL	GPIO port 2 output pull up enable	RW	0xFF
0x006C	GP2OCE	GPIO Port 2 open circuit enable	RW	0x00
0x0074	GP2IN <sup>2</sup>	GPIO port 2 data input	R	xx

<sup>1</sup> Contents of the GPxIN register depend on the digital level on the corresponding pins.

<sup>2</sup> Contents of the GPxIN register depend on the digital level on the corresponding pins.

GPIO Interface memory address table (continued)

Offset	Name	Description	Access	Default
0x0078	GP2OUT	GPIO port 2 data out	RW	0x00
0x007C	GP2SET	GPIO port 2 data out set	W	0x00
0x0080	GP2CLR	GPIO port 2 data out clear	W	0x00
0x0084	GP2TGL	GPIO port 2 pin toggle	W	0x00
0x0090	GP3CON	GPIO Port 3 configuration	RW	0x0000
0x0094	GP3OEN	GPIO port 3 output enable	RW	0x00
0x0098	GP3PUL	GPIO port 3 output pull up enable	RW	0xFF
0x009C	GP3OCE	GPIO Port 3 open circuit enable	RW	0x00
0x00A4	GP3IN <sup>1</sup>	GPIO port 3 data input	R	xx
0x00A8	GP3OUT	GPIO port 3 data out	RW	0x00
0x00AC	GP3SET	GPIO port 3 data out set	W	0x00
0x00B0	GP3CLR	GPIO port 3 data out clear	W	0x00
0x00B4	GP3TGL	GPIO port 3 data out toggle	W	0x00
0x00C0	GP4CON	GPIO Port 4 Configuration	RW	0x0000
0x00C4	GP4OEN	GPIO Port 4 output enable	RW	0x00
0x00C8	GP4PUL	GPIO Port 4 output pull up enable	RW	0xFF
0x00CC	GP4OCE	GPIO Port 4 open circuit enable	RW	0x00
0x00D4	GP4IN <sup>2</sup>	GPIO Port 4 data input	R	xx
0x00D8	GP4OUT	GPIO Port 4 data out	RW	0x00
0x00DC	GP4SET	GPIO Port 4 data out set	W	0x00
0x00E0	GP4CLR	GPIO Port 4 data out clear	W	0x00
0x00E4	GP4TGL	GPIO Port 4 data out toggle	W	0x00

---

<sup>1</sup> Contents of the GPxIN register depend on the digital level on the corresponding pins.

<sup>2</sup> Contents of the GPxIN register depend on the digital level on the corresponding pins.

**Table 217. Test Mode Access**

Address	Name	Description	Access	Default
0x40008824	RFTST	UHF transceiver test mode access	RW	0x0000

## GPIO Configurations Register

**Table 218: GPxCON Register Bit Description**

GP0CON Address: 0x40006000, GP1CON Address: 0x40006030, GP2CON Address: 0x40006060

GP3CON Address: 0x40006090, GP4CON Address: 0x400060C0

Bits	Name	Description
15 to 14	CON7	Configuration bits for Px.7 <sup>1</sup> as per Table 212
13 to 12	CON6	Configuration bits for Px.6 as per Table 212
11 to 10	CON5	Configuration bits for Px.5 as per Table 212
9 to 8	CON4	Configuration bits for Px.4 as per Table 212
7 to 6	CON3	Configuration bits for Px.3 as per Table 212
5 to 4	CON2	Configuration bits for Px.2 as per Table 212
3 to 2	CON1	Configuration bits for Px.1 as per Table 212
1 to 0	CON0	Configuration bits for Px.0 as per Table 212

## GPIO Output Enable Register

**Table 219: GPxOEN Register Bit Description**

GP0OEN Address: 0x40006004, GP1OEN Address: 0x40006034, GP2OEN Address: 0x40006064

GP3OEN Address: 0x40006094, GP4OEN Address: 0x400060C4

Bits	Name	Description
7 to 0 <sup>2</sup> (6 to 0)	OEN[7:0]  (OEN[6:0])	Output enable 1 – Enables the output on corresponding GPIO on port X 0 – Disables the output on corresponding GPIO on port X

<sup>1</sup> Not available in port 2

<sup>2</sup> 8 Bits in port 0, port 1, port 3, and port 4. Seven bits (0 to 6) in port 2.

## GPIO Pull Up Register

**Table 220: GPxPUL Register Bit Description**

GP0PUL Address: 0x40006008, GP1PUL Address: 0x40006038, GP2PUL Address: 0x40006068  
GP3PUL Address: 0x40006098, GP4PUL Address: 0x400060C8

Bits	Name	Description
7 to 0 <sup>1</sup> (6 to 0)	PUL[7:0]  (PUL[6:0])	Output enable 1 – Enables the internal pull up on corresponding GPIO on port X 0 – Disables the internal pull up on corresponding GPIO on port X

## GPIO Open Circuit Enable Register

**Table 221: GPxOCE Register Bit Description**

GP0OCE Address: 0x4000600C, GP1OCE Address: 0x4000603C, GP2OCE Address: 0x4000606C  
GP3OCE Address: 0x4000609C, GP4OCE Address: 0x400060CC

Bits	Name	Description
7 to 0 <sup>2</sup> (6 to 0)	OCE[7:0]  (OCE[6:0])	Output enable Sets the GPIO pads on port X to open circuit mode

## GPIO Data Input Register

**Table 222: GPxIN Register Bit Description**

GP0IN Address: 0x40006014, GP1IN Address: 0x40006044, GP2IN Address: 0x40006074  
GP3IN Address: 0x400060A4, GP4IN Address: 0x400060D4

Bits	Name	Description
7 to 0 <sup>3</sup> (6 to 0)	IN[7:0]  (IN[6:0])	Reflects the level on the GPIO pins except when in configured in open circuit.

<sup>1</sup> 8 Bits (0 to 7) in port 0, port 1, port 3, port 4, and port 5. Seven bits (0 to 6) in port 2.

<sup>2</sup> 8 Bits (0 to 7) in port 0, port 1, port 3, port 4, and port 5. Seven bits (0 to 6) in port 2.

<sup>3</sup> 8 Bits (0 to 7) in port 0, port 1, port 3, port 4, and port 5. Seven bits (0 to 6) in port 2.

## GPIO Data Out Register

**Table 223: GPxOUT Register Bit Description**

GP0OUT Address: 0x40006018, GP1OUT Address: 0x40006048, GP2OUT Address: 0x40006078  
GP3OUT Address: 0x400060A8, GP4OUT Address: 0x400060D8

Bits	Name	Description
7 to 0 <sup>1</sup> (6 to 0)	OUT [7:0]  (OUT [6:0])	Data out register. 1: Set by user code to drive the corresponding GPIO high. 0: Cleared by user to drive the corresponding GPIO low. Reads back the value on GPIO outputs. For example writing GP0OUT = 0x12 will drive 1 on P0.1 and P0.4 and the remaining GPIO will be low, assuming these pins are configured as outputs.

## GPIO Bit Set Register

**Table 224: GPxSET Register Bit Description**

GP0SET Address: 0x4000601C, GP1SET Address: 0x4000604C, GP2SET Address: 0x4000607C  
GP3SET Address: 0x400060AC, GP4SET Address: 0x400060DC

Bits	Name	Description
7 to 0 <sup>2</sup> (6 to 0)	SET [7:0]  (SET [6:0])	Data out register. 1: Set by user code to drive the corresponding GPIO High. 0: No action

## GPIO Bit Clear Register

**Table 225: GPxCLR Register Bit Description**

GP0CLR Address: 0x40006020, GP1CLR Address: 0x40006050, GP2CLR Address: 0x40006080  
GP3CLR Address: 0x400060B0, GP4CLR Address: 0x400060E0

Bits	Name	Description
7 to 0 <sup>3</sup> (6 to 0)	CLR [7:0]  (CLR [6:0])	Data out register. 1: Set by user code to drive the corresponding GPIO low. 0: Cleared by user code has no effect.

<sup>1</sup> 8 Bits (0 to 7) in port 0, port 1, port 3, port 4, and port 5. Seven bits (0 to 6) in port 2.

<sup>2</sup> 8 Bits (0 to 7) in port 0, port 1, port 3, port 4, and port 5. Seven bits (0 to 6) in port 2.

<sup>3</sup> 8 Bits (0 to 7) in port 0, port 1, port 3, port 4, and port 5. Seven bits (0 to 6) in port 2.

## GPIO Toggle Pin Register

**Table 226: GPxTGL Register Bit Description**

GP0TGL Address: 0x40006024, GP1TGL Address: 0x40006054, GP2TGL Address: 0x40006084  
GP3TGL Address: 0x400060B4, GP4TGL Address: 0x400060E4

Bits	Name	Description
7 to 0 <sup>1</sup> (6 to 0)	TGL [7:0] (TGL [6:0])	Toggle pin 1: Set by user code to invert the corresponding GPIO. 0: Cleared by user code has no effect.

## UHF Transceiver Test Mode Access

**Table 227. RFTST register bit description**

Address: 0x40008824

Bits	Name	Description
15 to 5	DIR	Controls the pin direction in digital test modes 1: Configure the GPIO as an output and reflects the state of the UHF transceiver pin. 0: The UHF transceiver pin is driven with the state on the external pin. RFTST[15] = DIR 10 RFTST[14] = DIR 9 : : : : RFTST[5] = DIR 0
4	Reserved	
3	AN1	Enable RF analog test mode Enables analog test mode, ATB 2, ATB 3 and ATB 4 on P1.6, P0.4 and P0.5. This mode has priority over the digital test mode.
2	AN0	Enable RF analog test mode Enables analog test mode ATB 1 on P1.5. This mode has priority over the digital test mode.
1	SPI0	Enable SPI0 out Connect the internal SPI signals from the UHF transceiver to P0.0, P0.1, P0.2 and P0.3.
0	GPX	Enable GPx out Connect the internal UHF transceiver GPIOs to external GPIO pins (P0.6, P0.7, P1.0, P1.1, P1.4, P1.5 and P1.6).

<sup>1</sup> 8 Bits (0 to 7) in port 0, port 1, port 3, port 4, and port 5. Seven bits (0 to 6) in port 2.



# I<sup>2</sup>C Serial Interface

## I<sup>2</sup>C Functionality

This block implements a master and slave I<sup>2</sup>C interface. I<sup>2</sup>C refers to a communications protocol originally developed by Philips Semiconductors (now NXP semiconductors)

### I<sup>2</sup>C Features:

- 2-byte transmit and receive FIFOs for the master and slave.
- Support for repeated starts.
- Support for 10 bit addressing.
- Master arbitration is supported.
- Continuous read mode for the master or up to 512 bytes fixed read.
- Clock stretching supported for the slave and the master.
- Support for 4 seven bit device addresses in the slave or 1 ten bit address and 2 seven bit addresses.
- Support for internal and external loop-back.
- Support for DMA.

The I<sup>2</sup>C bus peripheral has 2 pins used for data transfer. SCL is a serial clock and SDA is a serial data pin. The pins are configured in a Wired-AND format that allows arbitration in a multi-master system.

A master device can be configured to generate the serial clock. The frequency is programmed by the user in the serial clock divisor register. The master channel can be set to operate in Fast mode (400 kHz) or Standard mode (100 kHz).

The I<sup>2</sup>C bus peripheral's address in the I<sup>2</sup>C bus system is programmed by the user. This ID may be changed at any time while a transfer is not in progress. The user can set up to 4 slave addresses that will be recognized by the peripheral. The peripheral is implemented with a 2 byte FIFO for each transmit and receive shift register. IRQ pins and status bits in the control registers are available to signal to the processor core when the FIFO's need to be serviced.

## I<sup>2</sup>C Operation

The GPIOs used for I<sup>2</sup>C communication must be configured in I<sup>2</sup>C mode before enabling the I<sup>2</sup>C peripheral.

**Table 228: GPIO Port 1 Multiplex Table**

GPIO	Configuration Modes One (01)
P1.4	I2C0SCL
P1.5	I2C0SDA

### Master Transfer Initiation

If the Master enable bit (MASEN) is set, a master transfer sequence will be initiated by writing a value to the I2CADDRx register. If there is valid data in the I2CMTX register it will be the first byte transferred in the sequence after the address byte during a write sequence.

### Slave Transfer Initiation

If the Slave enable bit (SLVEN) is set in I2CSCON, a slave transfer sequence will be monitored for the device address in registers I2CID0, I2CID1, I2CID2, or I2CID3. If the device address is recognized then the part will participate in the slave transfer sequence as described in Figure 48 and Figure 49.

Note that a slave operation always starts with the assertion of one of 3 interrupt sources – Read Request (RXREQ), Write request (TXREQ) or General Call (GCINT) interrupt and the software should always look for a STOP interrupt to ensure the transaction has completed correctly and to de-assert the STOP interrupt status bit.

### RX/TX data FIFO's

The transmit data path for both master and slave consists of Tx FIFOs 2 bytes deep, I2CMTX and I2CSTX, and a transmit shifter. The Transmit Status bits in I2CMSTA [1:0] and I2CSSTA [0] denote whether there is valid data in the TX FIFO. Data from the TX FIFO is loaded into the TX shifter when a serial byte begins transmission. If the TX FIFO is not full during an active transfer sequence then the Transmit Request bit (TXREQ) in I2CMSTA or I2CSSTA will assert.

In the slave, if there is no valid data to transmit when the TX shifter is loaded, the Transmit Underrun status bit will assert.

The master will generate a STOP condition if there is no data in the transmit FIFO and the master is writing data (direction bit = 0).

The receive data path consists of a master and slave RX FIFO, each 2 bytes deep, I2CMRX and I2CSRX. The Receive request interrupt bits (RXREQ) in I2CMSTA or I2CSSTA indicate if there is valid data in the RX FIFO. Data is loaded into the RX FIFO after each byte is received.

If valid data in the RX FIFO is overwritten by the RX shifter, the Receive Overflow status bit will be asserted (I2CMSTA [9] or I2CSSTA [4]).

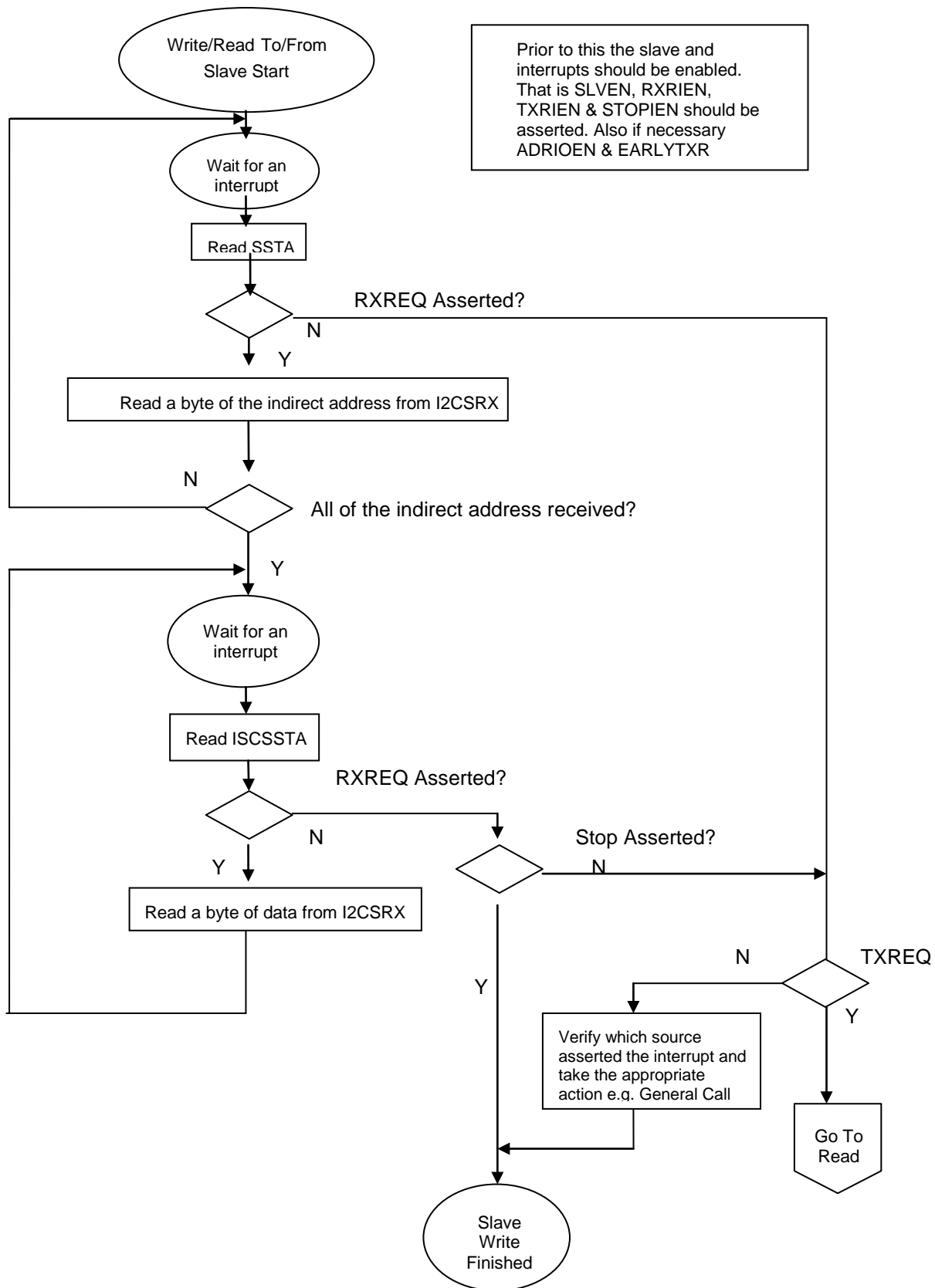
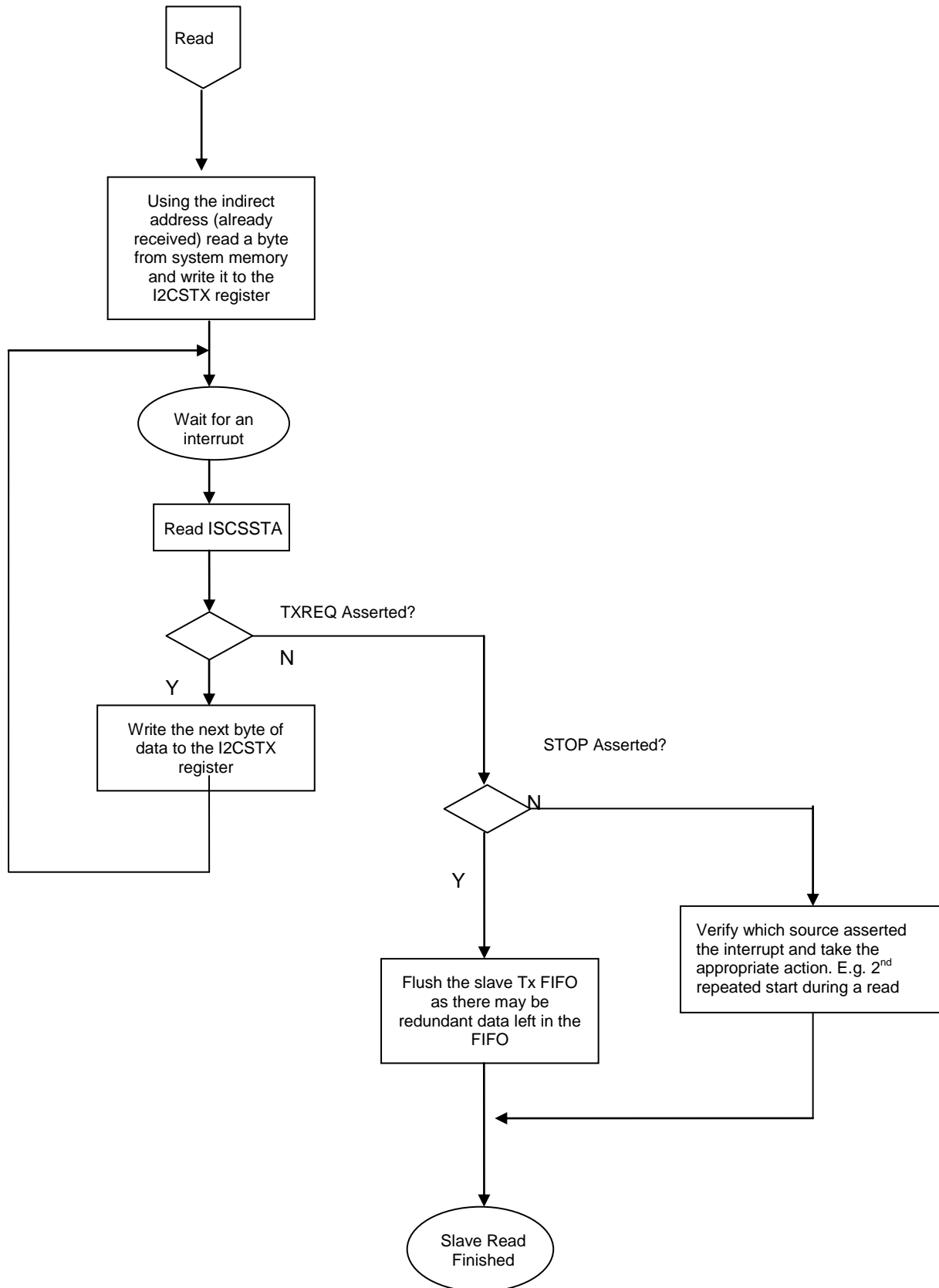


Figure 48: Slave read/write Flow Diagram

**Figure 49: Slave Read/Write Flow Diagram Continued – Read Portion**

### Master NOACK

When receiving data the master will respond with a NACK if its FIFO is full and an attempt is made to write another byte to the FIFO. This last byte received is not written to the FIFO and is lost.

### No acknowledge from the slave

If the slave does not want to acknowledge a read access then simply not writing data into the slave transmit FIFO will result in a NOACK.

If the slave does not want to acknowledge a master write then assert the NACK bit in the slave control register.

Normally the slave will ACK all bytes that are written into the receive FIFO. If the receive FIFO fills up the slave cannot write further bytes to it and it will not acknowledge the byte that it was not written to the FIFO. The master should then stop the transaction.

The slave will not acknowledge a matching device address if the direction bit is '1' (read) and the transmit FIFO is empty. Therefore there is very little time for the microcontroller to respond to a slave transmit request and the assertion of ACK. It is recommended that EARLYTXR is asserted for this reason.

### General Call

If the General call enable bit, GCEN (I2CSCON [2]) and the Slave enable bit, SLVEN, (I2CSCON [0]) are set the device will respond to a general call.

If the 2<sup>nd</sup> byte of the 'General Call' is 0x06 then the I<sup>2</sup>C interface (master and slave) is reset. The General Call interrupt status will assert and the General Call ID bits, GCID, (I2CSSTA [9:8]) will be 0x1. User code should take corrective action, which is, to reset the entire system or simply re-enable the I<sup>2</sup>C interface.

If the 2<sup>nd</sup> byte is 04h (Write programmable part of slave address by hardware), the general call interrupt status bit is asserted and the general call ID (GCID) will be 0x2.

The General Call interrupt status bit gets set on any General Call after the 2<sup>nd</sup> byte is received and user code should take corrective action, which is, to reprogram the device address, etc.

If GCEN is asserted the slave will always acknowledge the first byte of a 'General Call'. It will acknowledge the 2<sup>nd</sup> byte of a 'General Call' if the second byte is 0x04 or 0x06 or if the 2<sup>nd</sup> byte is a 'Hardware General Call' and HGCEN (I2CSCON [3]) is asserted.

The ALT register contains the alternate device ID for a 'Hardware general call' sequence. If the Hardware general call enable bit HGCEN, GCEN & SLVEN are all set, the device will recognize a 'Hardware General Call'. When a General call sequence is issued and the second byte of the sequence is identical to ALT, the Hardware call sequence is recognized for the device.

**Generation of repeated starts by the master**

The master will generate a repeated start if the I2CADR0 register is written while the master is still busy with a transaction. Once the state machine has started to transmit the device address it is then safe to write to the I2CADR0 register.

For instance if a 'write – repeated start – read/write' transaction is required then write to the I2CADR0 register after the state machine starts to transmit the device address or after the first TXREQ interrupt is received. When the transmit FIFO empties a repeated start will be generated.

Similarly if a 'read – repeated start – read/write' transaction is required then write to the '1<sup>st</sup> Master address byte' register after the state machine starts to transmit the device address or after the first RXREQ interrupt is received. When the requested 'receive count' is reached a repeated start will be generated.

**DMA Requests**

Four DMA channels are required to service the I2C master and slave. DMA enable bits are provided in the Slave control register and in the Master control register.

**I<sup>2</sup>C reset mode**

The slave state machine is reset when SLVEN is written to '0'.

The master state machine is reset when MASEN is written to '0'.

**I<sup>2</sup>C test modes**

The device can be placed in an internal loop-back mode by setting the LOOPBACK bit (I2CMCON [2]). There are 4 FIFOs (master Tx and Rx and slave Tx and Rx) so in effect the I<sup>2</sup>C peripheral can be setup to talk to itself. External loop-back can be performed if the master is setup to address the slave's address.

**I<sup>2</sup>C low power mode**

If the master and slave are both disabled (MASEN = SLVEN = 0) then the device is in its lowest power mode. That is the clocks to the master and slave state machines are both gated off.

## I<sup>2</sup>C Memory Mapped Registers

The I<sup>2</sup>C interface MMRs are based at address 0x40003000.

**Table 229: I<sup>2</sup>C Interface Memory Address Table Master Registers**

Base Address: 0x40003000

Offset	Name	Description	Access	Default
0x0000	I2CMCON	Master control register. Read write register.	RW	0x0000
0x0004	I2CMSTA	Master Status , Error, and IRQ register.	R	0x0000
0x0008	I2CMRX	Master receive data register.	R	0x0000
0x000C	I2CMTX	Master transmit data register.	W	0x0000
0x0010	I2CMRXCNT	Master receive data count register.	RW	0x0000
0x0014	I2CMCRXCNT	Master current receive data count register.	R	0x0000
0x0018	I2CADR1	1st Master Address Byte register.	RW	0x0000
0x001C	I2CADR2	2 <sup>nd</sup> Master Address Byte register (10 bit addresses only).	RW	0x0000
0x0020	I2CSBYT	Start byte register	RW	0x0000
0x0024	I2CDIV	Serial clock period divisor register.	RW	0x1F1F

## I<sup>2</sup>C Register Map Slave registers

**Table 230: I<sup>2</sup>C Interface Memory Address Table Slave Registers**

Base Address: 0x40003000

Offset	Name	Description	Access	Default
0x0028	I2CSCON	Slave control register	RW	0x0000
0x002C	I2CSSTA	Slave I2C Status, Error, and IRQ register.	R	0x0001
0x0030	I2CSRX	Slave receive data register.	R	0x0000
0x0034	I2CSTX	Slave transmit data register.	W	0x0000
0x0038	I2CALT	Hardware general call ID register.	RW	0x0000
0x003C	I2CID0	1 <sup>st</sup> slave address device ID.	RW	0x0000
0x0040	I2CID1	2 <sup>nd</sup> slave address device ID.	RW	0x0000
0x0044	I2CID2	3 <sup>rd</sup> slave address device ID.	RW	0x0000
0x0048	I2CID3	4 <sup>th</sup> slave address device ID.	RW	0x0000

## I<sup>2</sup>C Register Map Shared Registers

**Table 231: I<sup>2</sup>C Interface Memory Address Table Shared Registers**

Base Address: 0x40003000

Offset	Name	Description	Access	Default
0x004C	I2CFSTA	Master & slave RX & TX FIFO status register.	R/RW	0x0000
0x0050	I2CSHCON	Shared Control register	W	0x0000

I<sup>2</sup>C Master Control Register

Table 232: I2CMCON Register Bit Description

Address: 0x40003000

Bits	Name	Description
15 to 12	Reserved	Reserved bits
11	TXDMA	Enable master Tx DMA request. 1: Enable I2C master DMA requests. 0: Disable DMA mode.
10	RXDMA	Enable master Rx DMA request. 1: Enable I2C master DMA requests. 0: Disable DMA mode.
9	Reserved	Reserved
8	IENCMP	Transaction completed (or stop detected) interrupt enable. When asserted an interrupt is generated when a STOP is detected.
7	IENNACK	NACK received interrupt enable.
6	IENALOST	Arbitration lost interrupt enable.
5	IENTX	Transmit request interrupt enable.
4	IENRX	Receive request interrupt enable.
3	STRETCH	Stretch SCL enable. Setting this bit tells the device if SCL is '0' hold it at '0'. Or if SCL is '1' then when it next goes to '0' hold it at '0'.
2	LOOPBACK	Internal loop back enable. 1: SCL and SDA out of the device are muxed onto their corresponding inputs. Note that is also possible for the master to loop back a transfer to the slave as long as the device address corresponds, i.e. external loopback.
1	COMPETE	Start back-off disable. 1: Enables the device to compete for ownership even if another device is currently driving a START condition.
0	MASEN	Master enable 1: Enable Master 0: Master state machine flops are held in reset and the master is disabled. The master should be disabled when not in use as this will gate the clock to the master and save power. This bit should not be cleared until a transaction has completed, see the TCCOMP bit in the master status register. Note that APB writable register bits are not reset by this bit.



I<sup>2</sup>C Master Status Register

Table 233: I2CMSTA Register Bit Description

Address: 0x40003004

Bits	Name	Description
15 to 13	Reserved	Reserved bits. Returns 0 when read.
12	TXUR	Transmit FIFO Underrun. Set to 1 when the I <sup>2</sup> C master ends the transaction due to Tx-FIFO empty condition. This bit is only set when IENTX (I2CSCON[10]) is set.
11	MSTOP	STOP driven by I <sup>2</sup> C master Set to 1 when the I <sup>2</sup> C master drives a STOP condition on the I <sup>2</sup> C bus, therefore indicating a transaction completion, Tx- Underrun, Rx-overflow or a NACK by the slave. It is different from TCOMP because it is not set when the STOP condition occurs due to any other master on the I <sup>2</sup> C bus. This bit does not generate an interrupt. See TCOMP description below for available interrupts related to the STOP condition.
10	LINEBUSY	Line is busy. Set to 1 when a START is detected on the I <sup>2</sup> C bus. Cleared to 0 when a STOP is detected on the I <sup>2</sup> C bus.
9	RXOF	Receive FIFO overflow. Set to 1 when a byte is written to the receive FIFO when the FIFO is already full.
8	TCOMP	Transaction complete (or stop detected). (Can drive an interrupt.) Set to 1 when a STOP condition is detected on the I <sup>2</sup> C bus. If IENCOMP is '1', an interrupt will be generated when this bit asserts. This bit will only assert if the master is enabled (MASEN = 1). This bit should be used to determine when it is safe to disable the master. It can also be used to wait for another master's transaction to complete on the I2C bus when this master loses arbitration.
7	NACKDATA	NACK received in response to data write. (Can drive an interrupt.) Set to 1 when a NACK is received in response to a data write transfer. If IENNACK is '1', an interrupt will be generated when this bit asserts. Cleared on a read of the I2CMSTA register.
6	BUSY	Master busy. Set to 1 when the master state machine is servicing a transaction. Cleared to 0 if the state machine is idle or another device has control of the I <sup>2</sup> C bus.
5	ALOST	Arbitration lost. (Can drive an interrupt.) Set to 1 if the master loses arbitration. If IENALOST is '1', an interrupt will be generated when this bit asserts. Cleared to 0 on a read of the I2CMSTA register.
4	NACKADDR	NACK received in response to an address. (Can drive an interrupt.) Set to 1 if a NACK received in response to an address. If IENNACK is '1', an interrupt will be generated when this bit asserts. Cleared to 0 on a read of the I2CMSTA register.
3	RXREQ	Receive request. (Can drive an interrupt.) Set to 1 when there is data in the receive FIFO. If IENRX is '1', an interrupt will be generated when this bit asserts.

## I2CMSTA register bit description (Continued)

Bits	Name	Description
2	TXREQ	Transmit request. (Can drive an interrupt.) Set to 1 when the direction bit is '0' and the transmit FIFO is either empty or not full. If IENTX is '1', an interrupt will be generated when this bit asserts.
1 to 0	TXFSTA	Transmit FIFO status. 00: FIFO Empty. 01: Reserved 10: 1 byte in FIFO. 11: FIFO Full.

I<sup>2</sup>C Master Receive Register

Table 234: I2CMRX Register Bit Description

Address: 0x40003008

Bits	Name	Description
15 to 8	Reserved	Reserved bits. Returns 0 when read.
7 to 0	VALUE	Receive register. Read only. 0 by default. This register allows access to the receive data FIFO. The FIFO can hold 2 bytes.

I<sup>2</sup>C Master Transmit Register

Table 235: I2CMTX Register Bit Description

Address: 0x4000300C

Bits	Name	Description
15 to 8	Reserved	Reserved bits. Returns 0 when read.
7 to 0	VALUE	Transmit register. 0 by default. This register allows access to the transmit data FIFO. The FIFO can hold 2 bytes.

Note: I2CMTX is a write only register. If read the resulting values are undefined.

## I<sup>2</sup>C Master Receive Data Count Register

**Table 236: I2CMRXCNT Register Bit Description**

Address: 0x40003010

Bits	Name	Description
15 to 9	Reserved	Reserved bits. Returns 0 when read.
8	EXTEND	Extended read: Use this bit if greater than 256 bytes are required on a read. For example: To receive 412 bytes write 0x100 (EXTEND = 1) to the this register (I2CMRXCNT). Wait for the first byte to be received, then check the I2CMCRXCNT register for every byte received thereafter. When I2CMCRXCNT returns to '0', 256 bytes have been received. Then write 0x09C (412-256 = 156 decimal(0x9C) – with EXTEND = 0) to this register (I2CMRXCNT).
7 to 0	COUNT	Receive count. Program the number of bytes required minus one to this register. If just 1 byte is required write '0' to this register. If greater than 256 bytes are required, then use EXTEND.

## I<sup>2</sup>C Master Current Receive Count Register

**Table 237: I2CMCRXCNT Register Bit Description**

Address: 0x40003014

Bits	Name	Description
15 to 8	Reserved	Reserved bits. Returns 0 when read.
7 to 0	VALUE	Current receive count. This register gives the total number of bytes received so far. If 256 bytes are requested then this register will read '0' when the transaction has completed.

## I<sup>2</sup>C Master 1<sup>st</sup> Address byte Register

**Table 238: I2CADR1 Register Bit Description**

Address: 0x40003018

Bits	Name	Description
15 to 8	Reserved	Reserved bits. Returns 0 when read.
7 to 0	VALUE	Address byte. If a 7 bit address is required then I2CADR1[7:1] is programmed with the address and I2CADR1 [0] is programmed with the direction (read or write). If a 10 bit address is required then I2CADR1 [7:3] is programmed with '11110', I2CADR1 [2:1] is programmed with the 2 MSB's of the address and again I2CADR1 [0] is programmed with the direction (read or write).

## I<sup>2</sup>C Master 2<sup>nd</sup> Address Byte Register

**Table 239: I2CADR2 Register Bit Description**

Address: 0x4000301C

Bits	Name	Description
15 to 8	Reserved	Reserved bits. Returns 0 when read.
7 to 0	VALUE	Address byte. This register is only required when addressing a slave with 10 bit addressing. I2CADR2 [7:0] is programmed with the lower 8 bits of the address.

## I<sup>2</sup>C Start Byte Register

**Table 240: I2CSBYT Register Bit Description**

Address: 0x40003020

Bits	Name	Description
15 to 8	Reserved	Reserved bits. Returns 0 when read.
7 to 0	VALUE	Start byte. This register can be used to generate a start byte at the start of a transaction. To generate a start byte followed by a normal address, first write to VALUE ( in this register) and then write to the address register (I2CADR1). This will drive the byte written in VALUE on to the bus followed by a repeated start. This register can be used to drive any byte on to the I2C bus followed by a repeated start (not just a start byte '00000001').

## I<sup>2</sup>C Serial Clock Period Divider Register

**Table 241: I2CDIV Register Bit Description**

Address: 0x40003024

Bits	Name	Description
15 to 8	HIGH	Serial clock high time. This register controls the clock high time. The timer is driven by UCLK. Use the following equation to derive the required high time: $HIGH = (REQD\_HIGH\_TIME / UCLK\_PERIOD) - 2$ See example below. For example to generate a 400kHz SCL with a low time of 1250ns and a high time of 1250ns, with a UCLK frequency of 16MHz: $HIGH = 1250ns / 62.5ns - 2 = 0x12$ This register is reset to 0x1F which gives an SCL high time of 33 UCLK ticks.
7 to 0	LOW	Serial clock low time. This register controls the clock low time. The timer is driven by the core clock (UCLK). Use the following equation to derive the required low time. $LOW = (REQD\_LOW\_TIME / UCLK\_PERIOD) - 1$ For example to generate a 400kHz SCL with a low time of 1250ns and a high time of 1250ns, with a UCLK frequency of 16MHz: $LOW = 1250ns / 62.5ns - 1 = 0x13$ This register is reset to 0x1F which gives an SCL low time of 32 UCLK ticks.

I<sup>2</sup>C Slave Control Register

Table 242: I2CSCON Register Bit Description

Address: 0x40003028

Bits	Name	Description
15	Reserved	Reserved bit.
14	TXDMA	Enable slave Tx DMA request. 1: Enable I2C slave DMA requests. 0: Disable DMA mode.
13	RXDMA	Enable slave Rx DMA request. 1: Enable I2C slave DMA requests. 0: Disable DMA mode.
12	IENREPST	Repeated start interrupt enable. 1: Generate an interrupt when the REPSTART status bit asserts. 0: Disable an interrupt when the REPSTART status bit asserts.
11	Reserved	Reserved
10	IENTX	Transmit request interrupt enable. 1: Enable transmit request interrupt 0: Disable transmit request interrupt
9	IENRX	Receive request interrupt enable. 1: Enable receive request interrupt 0: Disable receive request interrupt
8	IENSTOP	Stop condition detected interrupt enable. 1: Enable stop condition detect interrupt. 0: Disable stop condition detect interrupt.
7	NACK	NACK next communication. 1: Allow the next communication to be NACK 'ed. This could be used for example if during a 24xx style access, an attempt was made to write to a 'read only' or non-existing location in system memory. That is the indirect address in a 24xx style write pointed to an un-writable memory location.
6	STRETCH	Stretch SCL enable. 1: Tell the device if SCL is '0' hold it at '0'. Or if SCL is '1' then when it next goes to '0' hold it at '0'.
5	EARLYTXR	Early transmit request mode. 1: Enable a transmit request just after the positive edge of the direction bit SCL clock pulse.
4	GCSBCLR	General call status bit clear. 1: Clear the General Call status & General Call ID bits The General Call status & General Call ID bits are not reset by anything other than a write to this bit or a full reset.

## I2CSCON register bit description (Continued)

Bits	Name	Description
3	HGCEN	<p>Hardware general call enable.</p> <p>When this bit and the General Call enable bit are set the device after receiving a general call, address 00h and a data byte checks the contents of the ALT against the receive shift register. If they match the device has received a 'hardware general call'. This is used if a device needs urgent attention from a master device without knowing which master it needs to turn to, it can use this call. This is a call "to whom it may concern". The device that requires attention embeds its own address into the message. The LSB of the ALT register should always be written to a 1.</p>
2	GCEN	<p>General call enable.</p> <p>1: Enable the I2C slave to ACK an I2C general call, address 0x00 (Write).</p>
1	ADR10EN	<p>Enable 10 bit addressing.</p> <p>1: Enable 10 bit addressing.</p> <p>One 10 bit address is supported by the slave and is stored in I2CID0 &amp; I2CID1, where I2CID0 contains the 1<sup>st</sup> byte of the address and the upper 5 bits must be programmed to '11110'. I2CID2 &amp; I2CID3 can be programmed with 7 bit addresses at the same time.</p> <p>0: If this bit is clear the slave can support 4 slave addresses, programmed in registers I2CID0 to I2CID3.</p>
0	SLVEN	<p>Slave enable.</p> <p>1: Enable Slave.</p> <p>0: Disable the slave and all slave state machine flops are held in reset</p> <p>Note that APB writable register bits are not reset.</p>

I<sup>2</sup>C Slave Status Register

Table 243: I2CSSTA Register Bit Description

Address: 0x4000302C

Bits	Name	Description
15	Reserved	Reserved bit. Returns 0 when read.
14	START	Start and matching address. Set to 1 if a 'START' is detected on SCL/SDA and a) The device address matched or b) A general call(GC - 0000_0000) code is received and GC is enabled or c) A High Speed (HS – 0000_1XXX) code is received or d) A start byte (0000_0001) is received. Cleared to 0 on receipt of either a STOP or START condition.
13	REPSTART	Repeated start and matching address. (Can drive an interrupt.) Set to 1 if START (I2CSSTA[14]) is already asserted and then a repeated start is detected. Cleared to 0 when read or on receipt of a STOP condition.
12 to 11	IDMAT	Device ID matched. Set to 00 when received Address matched ID register 0. Set to 01 when received Address matched ID register 1. Set to 10 when received Address matched ID register 2. Set to 11 when received Address matched ID register 3.
10	STOP	Stop after start and matching address. (Can drive an interrupt.) Set to 1 if the slave device received a STOP condition after a previous START condition and a matching address. Cleared to 0 by a read of the status register. If IENSTOP (I2CSCON[8]) is set then the slave interrupt request will assert when this bit is set.
9 to 8	GCID	General Call ID. Cleared when the GCSBCLR (I2CSCON[4]) is written to '1'. These status bits will not be cleared by a 'General call reset'. 00: No General call. 01: General Call Reset & Program address. 10: General Call Program address. 11: General Call matching alternative ID.
7	GCINT	General call interrupt. (Always drives an interrupt.) Set to 1 if the slave device receives a general call of any type. To clear write '1' to the I2CSCON[4]. If it was a general call reset, all registers will be at their default values. If it was a hardware general call the Rx FIFO holds the second byte of the general call and this can be compared with the ALT register.
6	BUSY	Slave busy. Set to 1 if the slave device receives an I2C START condition. Cleared to 0 by hardware on the following conditions: a) The address doesn't match an ID register. b) The slave device receives a I2C STOP condition. c) If a repeated start address doesn't match.

## I2CSSTA register bit description continued

Bits	Name	Description
5	NOACK	<p>NACK generated by the slave.</p> <p>Set to 1 to indicate that the slave responded to its device address with a NACK.</p> <p>Set under the following conditions:</p> <ul style="list-style-type: none"> <li>a) If there was no data to transmit and sequence was a slave read, the device address is NACK'ed.</li> <li>b) If the NACK bit was set in the slave control register and the device was addressed.</li> </ul> <p>Cleared to 0 on a read of the I2CSSTA register.</p>
4	RXOF	<p>Receive FIFO overflow.</p> <p>Set to 1 when a byte is written to the receive FIFO when the FIFO is already full.</p>
3	RXREQ	<p>Receive request. (Can drive an interrupt.)</p> <p>Set to 1 when the receive FIFO is not empty.</p> <p>Set on the falling edge of the SCL clock pulse that clocks in the last data bit of a byte.</p> <p>Cleared to 0 when the receive FIFO is read or flushed</p>
2	TXREQ	<p>Transmit request. (Can drive an interrupt.)</p> <p>If EARLYTXR = 0, TXREQ is set when the direction bit for a transfer is received high. There after, as long as the transmit FIFO is not full this bit will remain asserted. Initially it is asserted on the negative edge of the SCL pulse that clocks in the direction bit (if the device address matched also).</p> <p>If EARLYTXR = 1, TXREQ is set when the direction bit for a transfer is received high. There after, as long as the transmit FIFO is not full this bit will remain asserted. Initially it is asserted after the positive edge of the SCL pulse that clocks in the direction bit (if the device address matched also).</p> <p>This bit is cleared on a read of the I2CSSTA register.</p>
1	TXUR	<p>Transmit FIFO Underrun.</p> <p>Set to 1 if a master requests data from the device and the Tx FIFO is empty for the rising edge of SCL.</p>
0	TXFSEREQ	<p>Tx FIFO Status</p> <p>Set to 1 whenever the slave Tx FIFO is empty.</p>



## I<sup>2</sup>C Slave Receive Data Register

**Table 244: I2CSRX Register Bit Description**

Address: 0x40003030

Bits	Name	Description
15 to 8	Reserved	Reserved bit. Returns 0 when read.
7 to 0	VALUE	Receive register

## I<sup>2</sup>C Slave Transmit Data Register

**Table 245: I2CSTX Register Bit Description**

Address: 0x40003034

Bits	Name	Description
15 to 8	Reserved	Reserved bit. Returns 0 when read.
7 to 0	VALUE	Transmit register.

Note: I2CSTX is a write only register. If read the resulting values are undefined.

## I<sup>2</sup>C Slave Alt Register

**Table 246: I2CALT Register Bit Description**

Address: 0x40003038

Bits	Name	Description
15 to 8	Reserved	Reserved bit. Returns 0 when read.
7 to 0	VALUE	ALT register. This register is used in conjunction with HGCEN (I2CSCON[3]) to match a master generating a 'hardware general call'. It is used in the case where a master device cannot be programmed with a slave's address and instead the slave has to recognise the master's address.

## I<sup>2</sup>C Slave ID Registers

**Table 247: I2CIDx Register Bit Description**

I2CID0 Address: 0x4000303C

I2CID1 Address: 0x40003040

I2CID2 Address: 0x40003044

I2CID3 Address: 0x40003048

Bits	Name	Description
15 to 8	Reserved	Reserved bit. Returns 0 when read.
7 to 0	VALUE	There are 4 of these registers. I2CID0 to I2CID3. I2CID[7:1] is programmed with the device ID. I2CID[0] is don't care. See the I2CSCON[1] to see how these registers are programmed with a 10 bit address.

I<sup>2</sup>C FIFO Status Register

**Table 248: I2CFSTA Register Bit Description**  
**Address: 0x4000304C**

Bits	Name	Description
15 to 10	Reserved	Reserved bit. Returns 0 when read.
9	MFLUSH	Flush the master transmit FIFO. Writing a '1' to this bit flushes the master transmit FIFO. The master transmit FIFO will have to be flushed if arbitration is lost or a slave responds with a NACK.
8	SFLUSH	Flush the slave transmit FIFO. Writing a '1' to this bit flushes the slave transmit FIFO.
7 to 6	MRXFSTA	Master receive fifo status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty 01: 1 byte in the FIFO 10: 2 bytes in the FIFO 11: Reserved
5 to 4	MTXFSTA	Master transmit fifo status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty 01: 1 byte in the FIFO 10: 2 bytes in the FIFO 11: Reserved
3 to 2	SRXFSTA	Slave receive fifo status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty 01: 1 byte in the FIFO 10: 2 bytes in the FIFO 11: Reserved
1 to 0	STXFSTA	Slave transmit fifo status. The status is a count of the number of bytes in a FIFO. 00: FIFO empty 01: 1 byte in the FIFO 10: 2 bytes in the FIFO 11: Reserved

**Note on Tx FIFO status (master & slave):**

There is 1 shadow register outside the Tx FIFO's into which data is loaded before being transmitted. The Tx FIFO status can be setup to decrement when a byte is unloaded from the Tx FIFO into the shadow register or to decrement when the byte has been transmitted. The advantage of decrementing the FIFO status when it is unloaded from the FIFO is that a 'transmit interrupt' will assert earlier and the microcontroller has more time to respond to the interrupt and write a byte to the FIFO.

## I<sup>2</sup>C Shared Control Register

**Table 249: I2CSHCON Register Bit Description**  
Address: 0x40003050

Bits	Name	Description
15 to 1	Reserved	Reserved bit. Returns 0 when read.
0	RESET	<p>Reset START STOP detect circuit. Write only. Set this bit to reset the</p> <ol style="list-style-type: none"><li>1.SCL and SDA synchronisers and the</li><li>2.START &amp; STOP detect circuit and the</li><li>3.LINEBUSY detect circuit</li></ol> <p>These circuits are not reset when both the master and slave are disabled as LINEBUSY needs to assert even when the master is not enabled. It should only be necessary to reset these circuits after a power on reset in case SCL/SDA do not power up cleanly. Returns 0 when read.</p>

# Serial Peripheral Interfaces

## SPI Features

The ADuCRF101 integrates two complete and identical hardware serial peripheral interfaces (SPI). SPI is an industry standard, synchronous serial interface that allows eight bits of data to be synchronously transmitted and simultaneously received, that is, full duplex. The two SPIs implemented on the ADuCRF101 can operate to a maximum bit rate of 8Mb/s in both master and slave mode.

One of the SPI peripheral, SPI0, is dedicated to the UHF Transceiver communication and is not available to external pins.

The second SPI peripheral, SPI1, is available to the user to interface with external components. Both SPI's have additional DMA feature. It has two DMA channels that interface with  $\mu$ DMA controller of the ARM Cortex-M3. One DMA channel is used for transmit and the other for receive.

The standard SPI features included on both interfaces are:

- Serial clock phase mode and Serial clock polarity mode
- LSB first transfer option
- Loopback mode
- Master or slave mode
- Transfer and interrupt mode
- Continuous transfer mode
- Tx/Rx FIFO
- Interrupt mode, interrupt after 1, 2, 3 or 4 bytes
- Rx Overflow mode and Tx Underrun mode
- Open circuit data output mode

## SPI Operation

The SPI port can be configured for master or slave operation and consists of four pins: MISO, MOSI, SCL, and CS.

Note: The GPIOs used for SPI communication must be configured in SPI mode before enabling the SPI peripheral.

### MISO (Master In, Slave Out) Pin

The MISO pin is configured as an input line in master mode and an output line in slave mode. The MISO line on the master (data in) should be connected to the MISO line in the slave device (data out). The data is transferred as byte wide (8-bit) serial data, MSB first.

### MOSI (Master Out, Slave In) Pin

The MOSI pin is configured as an output line in master mode and an input line in slave mode. The MOSI line on the master (data out) should be connected to the MOSI line in the slave device (data in). The data is transferred as byte wide (8-bit) serial data, MSB first.

### SCL (Serial Clock I/O) Pin

The master serial clock (SCL) synchronizes the data being transmitted and received through the MOSI SCL period. Therefore, a byte is transmitted/received after eight SCL periods. The SCL pin is configured as an output in master mode and as an input in slave mode.

In master mode, polarity and phase of the clock are controlled by the SPIxCON register, and the bit rate is defined in the SPIxDIV register as follows:

$$f_{SERIALCLOCK} = \frac{UCLK}{2 \times (1 + SPIxDIV)}$$

The maximum data rate is 8Mbps.

In Slave mode, the SPIxCON register must be configured with the phase and polarity of the expected input clock. The slave accepts data from an external master up to 8 Mbps.

In both master and slave mode, data is transmitted on one edge of the SCL signal and sampled on the other. Therefore, it is important that the polarity and phase are configured the same for the master and slave devices.

### Chip Select ( $\overline{\text{CS}}$ Input) Pin

In SPI slave mode, a transfer is initiated by the assertion of  $\overline{\text{CS}}$ , which is an active low input signal. The SPI port then transmits and receives 8-bit data until the transfer is concluded by deassertion of  $\overline{\text{CS}}$ . In slave mode,  $\overline{\text{CS}}$  is always an input.

In SPI master mode, the  $\overline{\text{CS}}$  is an active low output signal. It asserts itself automatically at the beginning of a transfer and deasserts itself upon completion.

### SPI Transfer Initiation

In Master mode, the transfer and interrupt mode bit, SPIxCON [6] determines the manner in which an SPI serial transfer is initiated. If the mode bit is set then a serial transfer is initiated after a write to the TX FIFO occurs. If the mode bit is cleared then a serial transfer is initiated after a read of the RX FIFO, the read must be done while the SPI interface is idle. A read done during an active transfer will not initiate another transfer.

For any setting of SPIxCON [1] and SPIxCON [6], the SPI will simultaneously receive and transmit data. Therefore during data transmission the SPI is also receiving data and filling up the Rx FIFO. If the data is not read from the Rx FIFO the overflow interrupt will occur once the FIFO starts to overflow. If the user does not want to read the Rx data or receive overflow interrupts SPIxCON [12] can be set and the receive data will not be saved to the Rx FIFO.

Similarly when the user only wants to receive data and does not want to write data to the Tx FIFO SPIxCON [13] can be set to avoid getting Underrun interrupts from the Tx FIFO.

#### Tx Initiated Transfer

For transfers initiated by a write to the Tx FIFO the SPI will start transmitting as soon as the first byte is written to the FIFO irrespective of the configuration in SPIxCON [15:14]. The first byte is immediately read from the FIFO, written to the Tx shift register and the transfer commences.

If the Continuous transfer enable bit, SPIxCON [11], is set, the transfer will continue until no valid data is available in the Tx FIFO. There will be no stall period between transfers where Chip Select is de-asserted, Chip Select will be asserted and remain asserted for the duration of the transfer until Tx FIFO is empty. When the transfer stops does not depend on SPIxCON [15:14], the transfer will stop when there is no valid data left in the FIFO. Conversely the transfer will continue while there is valid data in the FIFO.

If the Continuous transfer enable bit SPIxCON [11] is cleared, each transfer consists of a single 8 bit serial transfer. If valid data exists in the Tx FIFO then a new transfer is initiated after a stall period where Chip Select is de-asserted.

#### RX Initiated Transfer

Transfers initiated by a read of the Rx FIFO depend on the number of bytes to be received in the FIFO. If SPIxCON [15:14] is set to 11 and a read to the Rx FIFO occurs then the SPI initiates a 4-byte transfer. If continuous mode is set the 4 bytes happen continuously with no de-assertion of Chip Select between bytes. If continuous mode is not set then the 4 bytes will happen with stall periods between transfers where the Chip Select will be de-asserted. A read of the Rx FIFO while the SPI is receiving data will not initiate another transfer after the present transfer is complete.

In Slave mode, a transfer is initiated by the assertion of the device's Chip Select.

Note that only CS0 is active in slave mode.

The device as a slave will transmit and receive 8 bit data until the transfer is concluded by the de-assertion of Chip Select.

The SPI Transfer Protocol diagrams, Figure 50 and Figure 51, illustrate the data transfer protocol for the SPI and the effects of CPHA and CPOL bits in the control register on that protocol. (CPOL is SPICPO in the diagram)

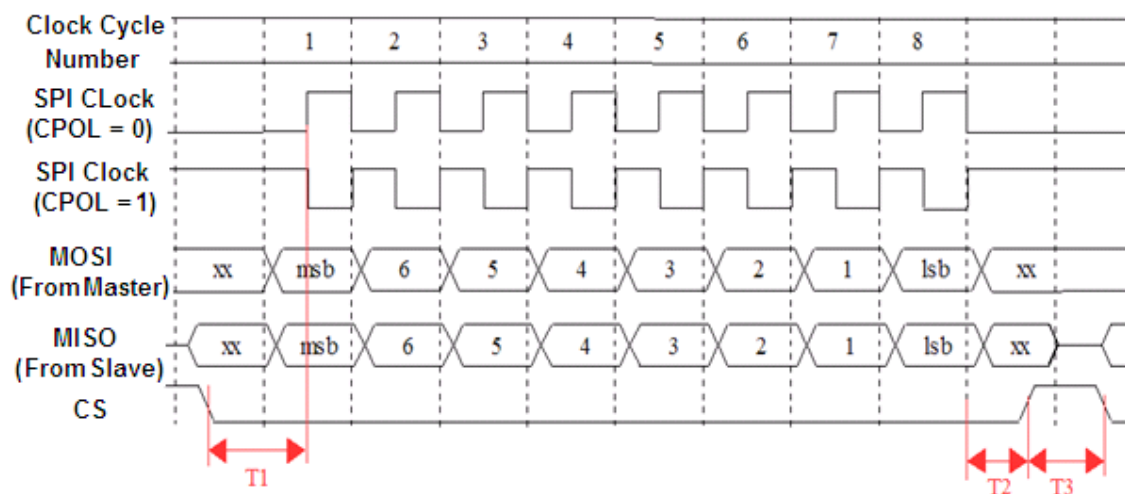


Figure 50: SPI Transfer Protocol CPHA = 0

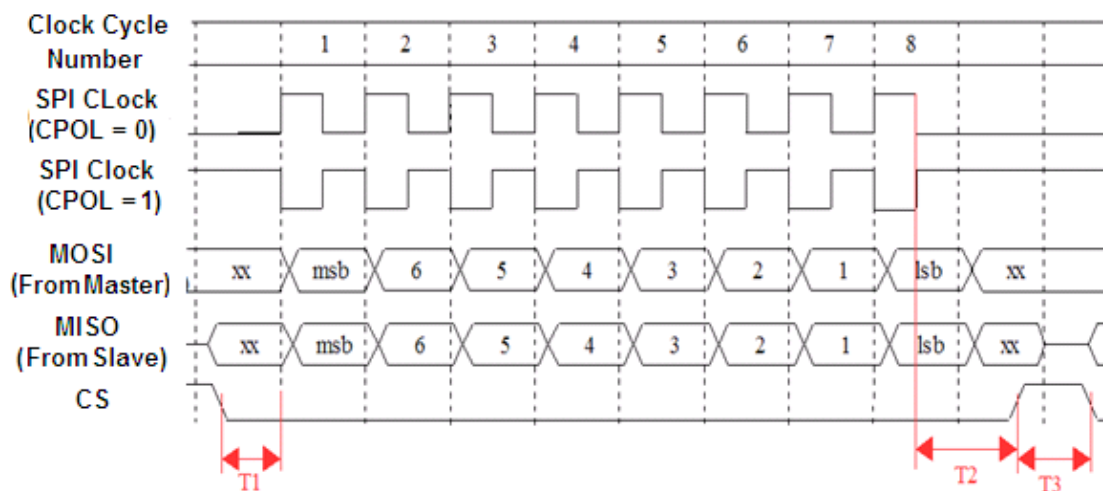


Figure 51: SPI Transfer Protocol CPHA = 1

### SPI Data Underrun and Overflow

If the Tx underrun mode bit, ZEN (SPIxCON [7]), is cleared, then the last 'stale' byte is shifted out when a transfer is initiated with no valid data in the FIFO. If ZEN is set, then zeros are transmitted when a transfer is initiated with no valid data in the FIFO.

If the Rx overflow overwrite enable bit, RXOF, is set then the valid data in the Rx FIFO is overwritten by the new serial byte received when there is no space left in the FIFO. If RXOF is cleared, then the new serial byte received is discarded when there is no space

left in the FIFO.

When valid data is being overwritten in the Rx FIFO the oldest byte is overwritten first followed by the next oldest byte and so on.

## SPI Interrupts

There is 1 interrupt line per SPI and 4 sources of interrupts. SPIxSTA [0] reflects the state of the interrupt line and SPIxSTA [7:4] the state of the 4 sources.

The SPI generates either TIRQ or RIRQ. Both interrupts cannot be enabled at the same time. The appropriate interrupt is enabled using the TIM bit in SPIxCON register. If TIM = 1, TIRQ is enabled. If TIM = 0, RIRQ is enabled.

### Tx Interrupt

If TIM is set, the Tx FIFO status causes the interrupt. SPIxCON [15:14] controls when the interrupt occurs. When SPIxCON [15:14] bits are set as follows:

**Table 250: SPIxCON[15:14] IRQ Mode Bits:**

SPIxCON[15:14]	Interrupt Condition
00	An interrupt is generated after each byte that is transmitted. The interrupt occurs when the byte is read from the FIFO and written to the shift register.
01	An interrupt is generated after every 2 bytes that are transmitted
10	An interrupt occurs after every third byte that is transmitted
11	An interrupt occurs after every fourth byte that is transmitted

The interrupts are generated depending on the number of bytes transmitted and not on the number of bytes in the FIFO. This is unlike the Rx interrupt which depends on the number of bytes in the Rx FIFO and not the number of bytes received.

The transmit interrupt is cleared by a read to the Status register. The status of this interrupt can be read by reading SPIxSTA [5]. The interrupt is disabled if SPIxCON [13] is left high.

NOTE: A write to the control register, SPIxCON, will reset the transmitted byte counter back to zero. For example, in a case where SPIxCON [15:14] is set to 11 and after 3 bytes have been transmitted SPIxCON is written to then the Tx interrupt will not occur until another 4 bytes have been transmitted.

### Rx Interrupt

If SPIxCON[6] is cleared the Rx FIFO status causes the interrupt. Again SPIxCON[15:14] controls when the interrupt will occur. The interrupt is cleared by a read of SPIxSTA. The status of this interrupt can be read by reading SPIxSTA[6].

Interrupts are only generated when data is written to the FIFO. For example if SPIxCON[15:14] is set to 00, an interrupt is generated after the first byte is received. When the Status register is read the interrupt is de-activated. If the byte is not read from the FIFO the interrupt will not be regenerated. Another interrupt will not be generated until another byte is received into the FIFO.



The interrupt depends on the number of valid bytes in FIFO and not the number of bytes received. For example when SPIxCON [15:14] is set to 01 an interrupt will be generated after a byte is received when there is 2 or more bytes in the FIFO. The interrupt is not generated after every 2 bytes received.

The interrupt is disabled if SPIxCON [12] is left high.

### Underrun/Overflow Interrupts

SPIxSTA [7] and SPIxSTA [4] also generate SPI interrupts.

When a transfer starts with no data in the TX FIFO SPIxSTA [4] gets set to indicate an Underrun condition. This will cause an interrupt. The interrupt (and Status bit) are cleared on a read of the Status register. This interrupt will occur irrespective of SPIxCON [15:14]. This interrupt is disabled if SPIxCON [13] is set.

When data is received when the RX FIFO is already full this will cause bit 8 of the Status register to go high indicating an overflow condition. This will cause an interrupt. The interrupt (and Status bit) are cleared on a read of the Status register. This interrupt will occur irrespective of SPIxCON [15:14]. This interrupt is disabled if SPIxCON [12] is set.

All interrupts are cleared by a read of the status register or if SPIxCON [0] is de-asserted. The Rx and Tx interrupts are also cleared if the relevant flush bits are asserted. Otherwise the interrupts will stay active even if the SPI is reconfigured.

### Wired-OR Mode (WOM)

In order to prevent contention when the SPI is used in multi-master or multi-slave system, data output pins MOSI and MISO can be configured to behave as open-circuit drivers. An external pull-up resistor is required when this feature is selected. The WOM bit (SPIxCON [4]) controls the pad enable outputs for the data lines.

### CSERR Condition

The SPI is reset after the completion of 8 clocks of SCLK. If there was an erroneous de-assertion of the CS signal before the completion of all the 8 SCLK cycles, then the bit\_counter stays in the value where it stopped and then it continues from there when CS gets asserted afterwards. This may cause inconsistent data transfers. To avoid this CSERR detection circuit is available. It checks for CS de-assertion when bit\_counter is not equal to 7 (reset value). If the condition is hit, it asserts a CSERR signal. Whenever CSERR occurs, there is a possibility that the final byte or set of bytes (based on irq\_mod) can get corrupted, irrespective of the bit\_cntr getting reset or not. It is strongly recommended that the enable, SPIxCON[0] is cleared after a CSERR to enable a clean recovery. This ensures that the subsequent transfers are error-free.

### SPI DMA

DMA operation is provided on both SPI's. 2 DMA channels are dedicated to transmit and receive. The SPI DMA channels should be configured in the uDMA controller of ARM.

It is possible to enable DMA request on 1 or 2 channels at the same time, by setting the DMA request bits for receive or transmit in the SPIxDMA register. If only the DMA transmit request (SPIxDMA [1]) is enabled, the RX FIFO will overflow during SPI transfer unless received data is read by user code, and an overflow interrupt will be generated. To avoid

generating overflow interrupts, the Rx FIFO flush bit should be set, or the SPI interrupt disabled in the NVIC. If only the DMA receive request (SPIxDMA [2]) is enabled, the Tx FIFO will be Underrun. Again to avoid Underrun interrupt the SPI interrupt should be disabled.

The SPI Tx (SPIxSTA [5]) and Rx (SPIxSTA [6]) interrupts are not generated when using DMA. The SPI TXUR (SPIxSTA [4]) and RXOF (SPIxSTA [7]) interrupts are generated when using DMA. SPIxCON [15:14] are not used in transmit mode and should be set to 00 in receive mode.

The ENABLE bit (SPIxDMA [0]) controls the start of a DMA transfer. DMA requests are only generated when ENABLE = 1. At the end of a DMA transfer i.e. when receiving a DMA SPI transfer interrupt, this bit needs to be cleared to prevent extra DMA requests to the uDMA controller. The data still present in the Tx FIFO will be transmitted if in Tx mode.

#### **DMA master transmit configuration:**

The DMA SPI Tx channel should be configured.

The NVIC should be configured to enable DMA Tx master interrupt

The SPI block should be configured as follow:

```
SPIxDIV = SPI_serial_freq;           //Configures serial clock frequency where
                                      $F_{\text{serial clock}} = f_{\text{uclk}} / (2 * (1 + \text{SPIDIV}))$ 
SPIxCON = 0x1043;                     //Enable SPI in master mode and transmit mode, Rx
FIFO flush enabled.
SPIxDMA = 0x3;                       //Enable DMA mode, enable Tx DMA request
```

When all data present in the DMA buffer are transmitted, the DMA generates an interrupt. User code should disable DMA request. Data will still be in the Tx FIFO as the DMA request is generated each time there is free space in the Tx FIFO, to keep the FIFO always full. User code can check how many bytes are still present in the FIFO in the FIFO status register.

#### **DMA master receive configuration:**

The SPIxCNT register is available in DMA receive master mode only. It sets the number of Receive bytes required by the SPI master, or the number of clocks that the master needs to generate. When the required number of bytes has been received, no more transfers are initiated. To initiate a DMA master receive transfer a dummy read should be done by user code. This dummy read should be added to the SPIxCNT number.

The counter counting the bytes as they are received is reset when SPI is disabled in SPIxCON [0], or if SPIxCNT register is modified by user code.

#### **To perform SPI DMA master receive:**

The DMA SPI Rx channel should be configured.

The NVIC should be configured to enable DMA Rx master interrupt.

The SPI block should be configured as follow:

```
SPIxDIV = SPI_serial_freq;           //Configures serial clock frequency where
                                      $F_{\text{serial clock}} = f_{\text{uclk}} / (2 * (1 + \text{SPIDIV}))$ 
SPIxCON = 0x2003;                     //Enable SPI in master mode and receive mode, 1
```

byte transfer.

SPIxDMA = 0x5; //Enable DMA mode, enable Rx DMA request

SPIxCNT = XXX; //Number of bytes to transferred + 1

A = SPIxRX; //Dummy read

The DMA transfer will stop when the number of clock has been generated. Note that the DMA buffer must be of the same size as SPIxCNT to generate a DMA interrupt when the transfer is complete.

## SPI and Power Down Modes

In master mode, before entering power down mode it is recommended to disable the SPI block in SPIxCON [0]. In slave mode, in either mode of operation, interrupt driven or DMA, the CS line level should be checked via the GPIO registers to ensure the SPI is not communicating and the SPI block should be disabled while the CS line is high. At power up the SPI block can be re-enabled.

## SPI Memory Mapped Registers

SPI0 interface consists of MMRs based at address 0x40004000.

SPI1 interface consists of MMRs based at address 0x40004400.

**Table 251: SPI Peripheral Memory Address Table**

SPI0 base address 0x40004000

SPI1 base address 0x40004400

Offset	Name	Description	Access	Default
0x0000	SPIxSTA <sup>1</sup>	Status register	R	0x0000
0x0004	SPIxRX	8-bit Receive register	R	0x0000
0x0008	SPIxTX	8-bit Transmit register	W	0x0000
0x000C	SPIxDIV	8-bit bit rate selection register	RW	0x0000
0x0010	SPIxCON	16-bit configuration register	RW	0x0000
0x0014	SPIxDMA	DMA enable register	RW	0x0000
0x0018	SPIxCNT	8-bit received byte count register	R	0x0000

<sup>1</sup> Where x is 0 or 1 for SPI0 or SPI1

## SPI Status Registers

**Table 252: SPIxSTA Register Bit Description**

**SPI0STA Address 0x40004000**

**SPI1STA Address 0x40004400**

Bits	Name	Description										
15 to 13		Reserved bits										
12	CSERR	Detected an abrupt CS de-assertion. Set to 1 when the CS line was de-asserted abruptly,even before the full-byte of data was transmitted completely. This bit will cause an interrupt. If the CSERR bit is set it recommended to clear the ENABLE bit in the SPIxCON register to ensure a clean recovery. Cleared to 0 when the SPISTA register is read.										
11	RXS	SPI Rx FIFO excess bytes present. Indicates when there is more bytes in the Rx FIFO than the Rx interrupt indicated. It depends on SPICON[15:14]:										
		<table><tr><th>SPICON[15:14]</th><th>RXS</th></tr><tr><td>00</td><td>RXS Set if there is 2 or more bytes in the RX FIFO.</td></tr><tr><td>01</td><td>RXS Set if there is 3 or more bytes in the RX FIFO.</td></tr><tr><td>10</td><td>RXS Set if there is 4 or more bytes in the RX FIFO.</td></tr><tr><td>11</td><td>RXS does not get set.</td></tr></table>	SPICON[15:14]	RXS	00	RXS Set if there is 2 or more bytes in the RX FIFO.	01	RXS Set if there is 3 or more bytes in the RX FIFO.	10	RXS Set if there is 4 or more bytes in the RX FIFO.	11	RXS does not get set.
		SPICON[15:14]	RXS									
		00	RXS Set if there is 2 or more bytes in the RX FIFO.									
		01	RXS Set if there is 3 or more bytes in the RX FIFO.									
10	RXS Set if there is 4 or more bytes in the RX FIFO.											
11	RXS does not get set.											

## SPIXSTA register bit description continued

Bits	Name	Description
4	TXUR	SPI Tx FIFO Underrun. (Interrupt) Set to 1 when a transmit is initiated without any valid data in the Tx FIFO. This bit generates an interrupt except when TFLUSH is set in SPICON. Cleared to 0 when the SPISTA register is read.
3 to 1	TXFSTA	Indicated how many valid bytes are in the SPI Tx FIFO: Set to 000 when Tx FIFO is empty Set to 001 when 1 valid byte in the FIFO Set to 010 when 2 valid bytes in the FIFO Set to 011 when 3 valid bytes in the FIFO Set to 100 when 4 valid bytes in the FIFO
0	IRQ	SPI Interrupt Status Bit. Set to 1 when an SPI based interrupt occurs. Cleared to 0 after reading SPISTA.

## SPI Receive Registers

Table 253: SPIxRX Register Bit Description

SPI0RX Address 0x40004004

SPI1RX Address 0x40004404

Bits	Name	Description
15 to 8	Reserved	These bits are Reserved and should be written 0 by user code.
7 to 0	VALUE	8-bit receive register A read of the RX FIFO will return the next byte to be read from the FIFO. A read of the FIFO when empty will return zeros.

## SPI Transmit Registers

Table 254: SPIxTX Register Bit Description

SPI0TX Address 0x40004004

SPI1TX Address 0x40004404

Bits	Name	Description
15 to 8	Reserved	These bits are Reserved and should be written 0 by user code.
7 to 0	VALUE	8-bit transmit register. A write to the Tx FIFO address space will write data to the next available location in the TX FIFO. If the FIFO is full the oldest byte of data in the FIFO will be overwritten. A read from this address location will return zeros.

## SPI Clock Divider Registers

**Table 255: SPIxDIV Register Bit Description**

SPI0DIV Address 0x4000400C

SPI1DIV Address 0x4000440C

Bits	Name	Description
15 to 7	Reserved	These bits are Reserved and should be written 0 by user code.
7	BCRST	Reset mode for CSERR. This bit is used to configure the expected behaviour of the SPI Interface logic after an abrupt de-assertion of CS: 1: SPI Interface logic get reset after a CSERR condition and the cortex is expected to clear the SPI ENABLE bit in SPIxCON 0: SPI Interface logic continues from where it stopped. The SPI can receive the remaining bits when CS gets asserted and cortex has to ignore the CSERR interrupt.
6	Reserved	Reserved
5 to 0	DIV	Factor used to divide UCLK to generate the serial clock $F_{\text{serial clock}} = f_{\text{uclk}} / (2^{(1+\text{SPIDIV})})$ Max frequency for serial clock is 1/2 UCLK These bits are only used for Master Mode. In slave mode there is no need to set the serial clock frequency. It will get the clock from the master.

Note: When setting the SPI serial clock the PCLK frequency must be taken into account. The PCLK frequency can be no less than half the SPI serial clock frequency.

For example:

SPI clock divide register is set to 0x0000 (SCL freq =  $\frac{1}{2}$  UCLK freq) then the maximum the CD bits can be set to is 2 (PCLK freq =  $\frac{1}{4}$  UCLK freq)

## SPI Control Registers

Table 256: SPIxCON Register Bit Description

SPI0CON Address 0x40004010

SPI1CON Address 0x40004410

Bits	Name	Description
15 to 14	MOD	<p>SPI IRQ mode bits. When TIM is set these bits configure when the Tx/Rx interrupts occur in a transfer. For DMA Rx transfer, these bits should be 00.</p> <p>00: Tx interrupt occurs when 1 byte has been transferred.  Rx interrupt occurs when 1 or more bytes have been received into the FIFO.</p> <p>01: Tx interrupt occurs when 2 bytes has been transferred.  Rx interrupt occurs when 2 or more bytes have been received into the FIFO.</p> <p>10: Tx interrupt occurs when 3 bytes has been transferred.  Rx interrupt occurs when 3 or more bytes have been received into the FIFO.</p> <p>11: Tx interrupt occurs when 4 bytes has been transferred.  Rx interrupt occurs when the Rx FIFO is full, or 4 bytes present.</p>
13	TFLUSH	<p>SPI Tx FIFO Flush enable bit.</p> <p>1: Flush the Tx FIFO. This bit does not clear itself and should be toggled if a single flush is required. If this bit is left high, then either the last transmitted value or "0x00" is transmitted depending on the ZEN bit. Any writes to the Tx FIFO are ignored while this bit is set.</p> <p>0: Disable Tx FIFO flushing.</p>
12	RFLUSH	<p>SPI Rx FIFO Flush enable bit.</p> <p>1: Flush the Rx FIFO. This bit does not clear itself and should be toggled if a single flush is required. If this bit is set all incoming data is ignored and no interrupts are generated. If set and TIM = 0, a read of the Rx FIFO will initiate a transfer.</p> <p>0: Disable Rx FIFO flushing.</p>
11	CON	<p>Continuous Transfer Enable.</p> <p>1: Enable continuous transfer.</p> <p>In master mode, the transfer continues until no valid data is available in the Tx register. CS is asserted and remains asserted for the duration of each 8-bit serial transfer until Tx is empty.</p> <p>0: Disable continuous transfer. Each transfer consists of a single 8-bit serial transfer. If valid data exists in the SPITX register, then a new transfer is initiated after a stall period of 1 serial clock cycle. The CS line is de-activated for this 1 serial clock cycle.</p>
10	LOOPBACK	<p>Loop back Enable bit.</p> <p>1: connect MISO to MOSI thus data transmitted from Tx register is looped back to the Rx register. The MASEN bit must be set for loop back mode to work.</p> <p>0: normal mode.</p>
9	OEN	<p>Slave MISO Output enable bit.</p> <p>1: MISO to operate as normal.</p> <p>0: Disable the output driver on the MISO pin. The MISO pin will be Open-Circuit when this bit is clear.</p>
8	RXOF	<p>SPIRX Overflow Overwrite Enable.</p> <p>1: The valid data in the Rx register is overwritten by the new serial byte received.</p> <p>0: The new serial byte received is discarded.</p>
7	ZEN	<p>SPI transmit Zeros when Tx FIFO is empty.</p> <p>1: Transmit "0x00" when there is no valid data in the Tx FIFO.</p> <p>0: Transmit the last transmitted value when there is no valid data in the Tx FIFO.</p>

## SPIxCON register bit description continued

Bits	Name	Description
6	TIM	SPI Transfer and Interrupt Mode. 1: Initiate transfer with a write to the SPI TX register. Interrupt only occurs when Tx is empty. 0: Initiate transfer with a read of the SPI RX register. The read must be done while the SPI interface is idle. Interrupt only occurs when Rx is full.
5	LSB	LSB First Transfer Enable Bit. 1: LSB is transmitted first. 0: MSB is transmitted first.
4	WOM	SPI Wired Or Mode enable bit. 1: Enable Open circuit data Output enable. <u>Master Mode:</u> When a 0 is being transmitted on the MOSI pin the output driver is enabled. When a 1 is being transmitted on MOSI pin the output driver is disabled and an external pull up resistor is required to pull the pin high. Typical resistor value 1kOhms. <u>Slave Mode:</u> When a 0 is being transmitted on the MISO pin the output driver is enabled. When a 1 is being transmitted on MISO pin the output driver is disabled and an external pull up resistor is required to pull the pin high. Typical resistor value 1kOhms.  0: Normal output levels.
3	CPOL	Serial Clock Polarity Mode Bit. 1: Serial clock idles high. 0: Serial clock idles low.
2	CPHA	Serial Clock Phase Mode Bit. 1: Serial clock pulses at the start of the first data bit 0: Serial clock pulses at the middle of the first data bit transfer.
1	MASEN	Master mode enable bit 1: Enable master mode. 0: Enable slave mode.
0	ENABLE	SPI Enable bit. 1: Enable the SPI. 0: Disable the SPI. Clearing this bit will also reset all the FIFO related logic and bit_counter to enable a clean start.

## Note:

1. When changing the configuration care must be taken not to change it during a data transfer to avoid corrupting the data. It is recommended to change configuration when the module is disabled ( disable the SPI, ENABLE =0) , then reconfigure and then re-enable the SPI (ENABLE = 1)
2. When reconfiguring from slave mode to master mode or visa-versa both FIFOs must be empty
3. When using DMA, MOD (SPIxCON [15:14]) settings become irrelevant.



4. Though the interrupt generation logic is independent of MOD (SPIxCON [15:14]), the master transfer initiation logic still depends on MOD. However the DMA service happens only in a byte-by-byte basis and not in bursts. Therefore only the value of MOD = 0 should be used in the case of DMA.

## SPI DMA Enable Registers

**Table 257: SPIxDMA Register Bit Description**

SPI0DMA Address 0x40004014

SPI1DMA Address 0x40004414

Bits	Name	Description
15 to 3	Reserved	These bits are Reserved and should be written 0 by user code.
2	IENRXDMA	1: Receive DMA request. Cleared by default.
1	IENTXDMA	1: Enable transmit DMA request. Cleared by default.
0	ENABLE	1: Start a DMA transfer 0: By user at the end of DMA transfer. This bit needs to be cleared to prevent extra DMA request to the uDMA controller.

**Note:**

1. DMA requests are not generated when DMA ENABLE = 0
2. At the end of a DMA transfer this bit needs to be reset to prevent extra DMA requests to the uDMA controller.
3. When ENABLE (DMA) = 1, Tx (SPIxSTA [5]) and Rx (SPIxSTA [6]) interrupts are automatically disabled. However the TXUR (SPIxSTA [4]) and RXOF (SPIxSTA [7]) interrupts are still available to indicate TX FIFO underrun and RX FIFO overflow respectively.

## SPI Received Byte Count Registers

**Table 258: SPIxCNT Register Bit Description**

SPI0CNT Address 0x40004018

SPI1CNT Address 0x40004418

Bits	Name	Description
15 to 8	Reserved	These bits are Reserved and should be written 0 by user code.
7 to 0	VALUE	Number of RX bytes required in master mode. Disabled in TX mode. When the required number of bytes are received no more transfers are initiated.

# UART Serial Interface

## UART Features

The UART peripheral is a full-duplex Universal Asynchronous Receiver / Transmitter, compatible with the industry standard 16450. The UART is responsible for converting data between serial and parallel formats. The serial communication follows an asynchronous protocol supporting various word length, stop bits, and parity generation options.

This UART also contains modem control and interrupt handling hardware. The UART features a fractional divider that facilitates high accuracy baud rate generation.

Interrupts may be generated from a number of unique events such as data buffer full/empty, transfer error detection and break detection.

## UART Operation

### Serial Communications

An asynchronous serial communication protocol is followed with the following options:

5 – 8 data bits

1, 2 or 1 & ½ stop bits

None, even or odd parity

Baud rate =  $UCLK \div (2 \times 16 \times COMDIV) \div (M + N \div 2048)$

Where COMDIV = 1 to 65536, M = 1 to 3 and N = 0 to 2047.

All data words require a start bit and at least one stop bit. This creates a range from 7 – 12 bits for each word. Transmit operation is initiated by writing to the Transmit Holding Register (COMTX). After a synchronization delay the data is moved to the Transmit Shift Register (TSR) where it will be shifted out at a baud (bit) rate equal to  $UCLK \div (2 \times 16 \times COMDIV) \div (M + N \div 2048)$  with start, stop, and parity bits appended as required. All data words begin with a low going start bit. The transfer of the COMTX to the TSR causes the Transmit Register Empty Status flag to be set.

Receive operation uses the same data format as the transmit configuration except for the number of stop bits which are always one. After detection of the start bit, the received word is shifted in the Receive Shift Register (RSR). After the appropriate number of bits (including stop bits) are received the data and any status is updated and the RSR is transferred to the Receive Buffer Register (COMRX). The Receive Buffer Register Full status flag is updated upon the transfer of the received word to this buffer and the appropriate synchronization delay.

A sampling clock equal to 16 times the baud rate is used to sample the data as close to the midpoint of the bit as possible. A receive filter is also present that removes spurious pulses of less than two times the sampling clock period.

Note that data is transmitted and received Least Significant Bit first (i.e. TSR [0]). This is often not the assumed case by the user. This however is standard for the protocol.

## Programmed IO Mode

In this mode the software is responsible for moving data to and from the UART. This is typically accomplished by interrupt service routines that respond to the transmit and receive interrupts by either reading or writing data as appropriate. This mode puts certain constraints on the software itself in that the software must respond within a certain time in order to prevent overflow errors from occurring in the receive channel.

Programmed IO mode also includes polling the status flags to determine when it is okay to move data.

Polling the status flag is processor intensive and not typically used unless the system can tolerate the overhead. Interrupts can be disabled using the COMIEN register.

Writing the COMTX when it is not empty or reading the COMRX when it is not full produce incorrect results and should not be done. In the former case the COMTX is overwritten by the new word and the previous word is never transmitted, in the latter case the previously received word will be read again. Both of these errors must be avoided in software by correctly using either interrupts or the status register polling. These errors are not detected in hardware.

## DMA Mode

In this mode, user code does not move data to and from the UART. DMA request signals going to external DMA block indicate that the UART is ready to transmit or receive data. These DMA request signals can be disabled in the COMIEN register.

## Enable/Disable Bit

Before the ADuCRF101 enters power down mode, it is recommended to disable the serial interfaces. A bit is provided in the UART control register to disable the UART serial peripheral. This bit disables the clock to the peripheral. When setting this bit, care must be taken in software that no data are being transmitted or received. If set during communication the data transfer will not complete: the receive or transmit register will contain only part of the data.

## Interrupts

The UART peripheral has 1 interrupt output to the interrupt controller for both RX and TX interrupts. The COMIIR register must be read by software to determine the cause of the interrupt. Note that in DMA mode, the Break and Modem status interrupts are not available.

In IO Mode, when receiving, the interrupt is generated for the following cases:

- COMRX Full
- Receive Overflow Error
- Receive Parity Error
- Receive Framing Error
- Break Interrupt (UART input (RXSIN) held low)
- Modem status interrupt (changes to CTS)
- COMTX Empty

## Buffer Requirements

This UART is double buffered (holding register and shift register).

## UART Memory Mapped Registers

The UART interface consists of MMRs based at address 0x40005000.

**Table 259. UART Interface Memory Address Table**

Base Address 0x40005000

Offset	Name	Description	Access	Default
0x0000	COMTX	Transmit holding register	W	0x0000
0x0000	COMRX	Receive buffer register	R	0x0000
0x0004	COMIEN	Interrupt enable register	RW	0x0000
0x0008	COMIIR	Interrupt Identification Register	R	0x0001
0x000C	COMLCR	Line Control Register	RW	0x0000
0x0010	COMMCR	Module Control Register	RW	0x0000
0x0014	COMLSR	Line status register	R	0x0060
0x0018	COMMSR	Modem status register	R	0x0000
0x0024	COMFBR	Fractional baudrate register	RW	0x0000
0x0028	COMDIV	Baudrate divider register	RW	0x0001
0x0030	COMCON	Control register	RW	0x0000

## UART Transmit and Receive Registers

COMRX and COMTX share the same address while they are implemented as different registers. If written to, then user accesses the Transmit Holding Register (COMTX), if read, user accesses the receive buffer register (COMRX).

COMRX: This is an 8-bit register that the user can read received data from. If the ERBFI bit is set in COMIEN register, then an interrupt is generated when this register is fully loaded with the received data via serial input port.

Note when user set the ERBFI bit when COMRX is already full, an interrupt is generated immediately.

COMTX: This is an 8-bit register that the user can write to with the data to be sent. If ETBEI bit is set in COMIEN register, an interrupt will be generated when COMTX is empty.

Note that if you set ETBEI while COMTX is already empty, an interrupt will be generated immediately.

**Table 260: COMRX/COMTX Register Bit Description**

Address 0x40005000

Bits	Name	Description
7 to 0	VALUE	Receive Buffer Register / Transmit Holding Register

## UART Interrupt Enable Register

COMIEN is the interrupt enable register which is used to configure which interrupt source will generate interrupt. Only the lowest 4 bits in this register enable interrupt. Bit 4 and bit 5 enable UART DMA signals. The UART DMA channel and interrupt must be configured in the DMA block.

**Table 261: COMIEN Register Bit Description**

Address 0x40005004

Bits	Name	Description
7 to 6	Reserved	Reserved
5	EDMAR	DMA requests in transmit mode. 1: Enable 0: Disable
4	EDMAT	DMA requests in receive mode. 1: Enable 0: Disable
3	EDSSI	Modem Status interrupt (This interrupt is generated when any of MSR[3:0] is set) 1: Enable 0: Disable
2	ELSI	Rx status interrupt (This interrupt is generated when any bit of LSR[4:1] is set) 1: Enable 0: Disable
1	ETBEI	Transmit buffer empty interrupt 1: Enable 0: Disable
0	ERBFI	Receive buffer full interrupt 1: Enable 0: Disable

## UART Interrupt Identification Register

**Table 262: COMIIR Register Bit Description**

Address 0x40005008

Bits	Name	Description
7 to 3	Reserved	Reserved
2 to 1	STA	Status bits. Status bits are used to encode the interrupt status when NIRQ is low. See Table 263 for more details.
0	NOTIRQ	Interrupt flag. 1: There is no interrupt (default) 0: Indicates any of the following: a) receive buffer full b) transmit buffer empty c) line status d) modem status interrupt occurs

**Table 263: Interrupt Identification Table**

Bit 2:1 STA	Bit 0 NIRQ	Priority	Definition	Clearing Operation
00	1	-	No interrupt	-
11	0	1	Receive line status interrupt	Read COMLSR register
10	0	2	Receive buffer full interrupt	Read COMRX register
01	0	3	Transmit buffer empty interrupt	Write data to COMTX or read COMIIR
00	0	4	Modem status interrupt	Read COMMSR register

## UART Line Control Register

**Table 264: COMLCR Register Bit Description**

Address 0x4000500C

Bits	Name	Description
7	Reserved	Reserved
6	BRK	Set Break. Set by user to force TxD to 0. Cleared to operate in normal mode.
5	SP	Stick Parity. 1: To force parity to defined values based on EPS and PEN values. EPS = 1 and PEN = 1, Parity forced to 1 EPS = 0 and PEN = 1, Parity forced to 0 EPS = X and PEN = 0, No parity transmitted 0: Parity will not be forced based on EPS and PEN values
4	EPS	Even Parity Select Bit. 1: Even parity. 0: Odd parity.
3	PEN	Parity Enable Bit. 1: transmit and check the parity bit. 0: No parity transmission or checking.
2	STOP	Stop Bit. 1: Transmit 1.5 stop bits if the word length is 5 bits, or 2 stop bits if the word length is 6, 7, or 8 bits. The receiver checks the first stop bit only, regardless of the number of stop bits selected. 0: Generate one stop bit in the transmitted data.
1 to 0	WLS	Word Length Select bits 00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits

## UART Modem Control Register

**Table 265: COMMCr Register Bit Description**  
Address 0x40005010

Bits	Name	Description
7 to 5	Reserved	Reserved
4	LOOPBACK	Loop Back. 1: Enable loop back mode. In loop back mode, the UART TXD is forced high. The modem signals are also directly connected to the status inputs (RTS to CTS). 0: Normal mode.
3	OUT1	Parity Enable Bit. 1: Transmit and check the parity bit. 0: no parity transmission or checking.
2	OUT2	Stop Bit. 1: Transmit 1.5 stop bits if the word length is 5 bits or 2 stop bits if the word length is 6 bits, 7 bits, or 8 bits. The receiver checks the first stop bit only, regardless of the number of stop bits selected. 0: Generate 1 stop bit in the transmitted data.
1	RTS	Request To Send. 1: Force the RTS output to 0. 0: Force the RTS output to 1.
0	Reserved	Reserved

## UART Line Status Register

**Table 266: COMLSR Register Bit Description**  
Address 0x40005014

Bits	Name	Description
7	Reserved	Reserved
6	TEMT	COMTX and Shift Register Empty Status Bit. 1: If COMTX and the shift register are empty. This bit indicates that the data has been transmitted, that is, it is no longer present in the shift register. (Default) 0: Cleared when writing to COMTX.
5	THRE	COMTX Empty Status Bit. 1: If COMTX is empty. COMTX can be written as soon as this bit is set, the previous data may not have been transmitted yet and can still be present in the shift register. (Default) 0: When writing to COMTX.
4	BI	Break Indicator. 1: When UART RXD is held low for more than the maximum word length. Cleared automatically.



## COMLSR Register Bit Description continued

Bits	Name	Description
3	FE	Framing Error. 1: When the stop bit is invalid. Cleared automatically.
2	PE	Parity Error. 1: When a parity error occurs. Cleared automatically.
1	OE	Overrun Error. 1: Automatically if data are overwritten before being read. Cleared automatically.
0	DR	Data Ready. 1: Automatically when COMRX is full. Cleared by reading COMRX.

## UART Modem Status Register

Table 267: COMMSR Register Bit Description

Address 0x40005018

Bits	Name	Description
7 to 5	Reserved	Reserved
4	CTS	Clear To Send 1 when CTS is currently logic low 0 when CTS is currently logic high
3 to 1	Reserved	Reserved
0	DCTS	Delta CTS. 1: Automatically if CTS changed state since COMMSR last read. Cleared automatically by reading COMMSR.

## UART Fractional Baud Rate Divider Register

**Table 268: COMFBR Register Bit Description**  
Address 0x40005024

Bits	Name	Description
15	ENABLE	<p>Fractional baud rate generator enable bit for more accurate baud rate generation. The generating of a fractional baud rate can be described by the following formula and the final baud rate of UART operation can be shown in the diagram below.</p> $Baudrate = \frac{UCLK}{2 \times (M + N / 2048)} \times 16 \times COMDIV$
14 to 13	Reserved	Reserved
12 to 11	DIVM	Fractional baud rate M divide bits (1 to 3). This bit should not be 0.
10 to 0	DIVN	Fractional baud rate N divide bits (0 to 2047).

**Table 269: Baudrate Examples**

Baudrates	COMDIV	DIVM	DIVN	Actual	% Error
9600	17	3	131	9599.25	-0.0078%
19200	8	3	523	19199.04	-0.0050%
38400	4	3	523	38398.08	-0.0050%
57600	8	1	174	57605.76	0.0100%
115200	2	2	348	115211.5	0.0100%
230400	2	1	174	230423	0.0100%
460800	1	1	174	460846.1	0.0100%

## UART Divider Register

The Baud rate divider register is a 16-bit register used to generate the baud rate for UART data transfer. The baud rate without fractional divider is a divided down version of master clock as shown below:

$$\text{Baud rate} = \text{UCLK} \div (2 \times 16 \times \text{COMDIV}) \div (\text{M} + \text{N} \div 2048)$$

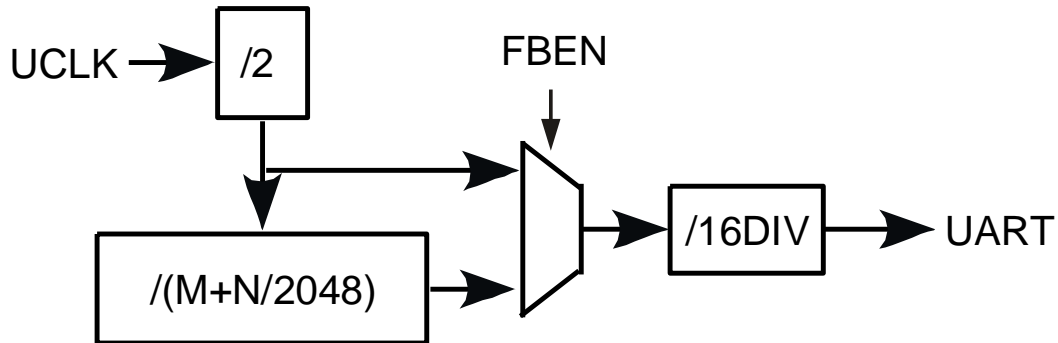


Figure 52: Baudrate Generation

Table 270: COMDIV Register Bit Description

Address 0x40005028

Bits	Name	Description
15 to 0	VALUE	Set the baudrate. The COMDIV register should not be 0.

## UART Control Register

Table 271: COMCON Register Bit Description

Address 0x40005030

Bits	Name	Description
15 to 1	Reserved	Reserved
0	ENABLE	UART disable bit. 1: Disable the UART. This reset the state machine and internal counters but the contents of the MMRs remain unchanged. It is advised to disable the UART before entering power down mode. 0: Enable the UART.

# General Purpose Timers

## General Purpose Timers Features

Timer 0 and timer 1 are two identical general-purpose 16-bit count-up/count-down timers. They can be clocked from four different clock sources: 32kHz internal oscillator (LFOSC), external crystal (LFXTAL), UCLK or PCLK. This clock source can be scaled down using a prescaler of 1, 16, 256 or 32,768.

The timers can be either free-running or periodic. In free-running mode, the counter decrements/increments from the maximum/minimum value until zero/full scale and starts again at the maximum/minimum value. In periodic mode, the counter decrements/increments from the value in the load register (TxLD MMR) until zero/full scale and starts again at the value stored in the load register.

The value of a counter can be read at any time by accessing its value register (TxVAL). This register is synchronous to UCLK. Therefore when a timer is clocked from a clock other than UCLK clock TxVAL reflects the latest timer value.

Timers are started by writing in the control register of the corresponding timer (TxCON).

An IRQ is generated each time the value of the counter reaches zero when counting down, or each time the counter value reaches full scale when counting up. An IRQ can be cleared by writing 1 to the time clear interrupt register of that particular timer (TxCLRI).

In addition, timer0 and timer1 have a capture register (TxCAP) that is triggered by a selected IRQ source initial assertion. When triggered, the current timer value is copied to TxCAP, and the timer continues to run. This feature can be used to determine the assertion of an event with increased accuracy. Each timer can capture 16 different events, listed in Table 272.

## General Purpose Timers Block Diagram

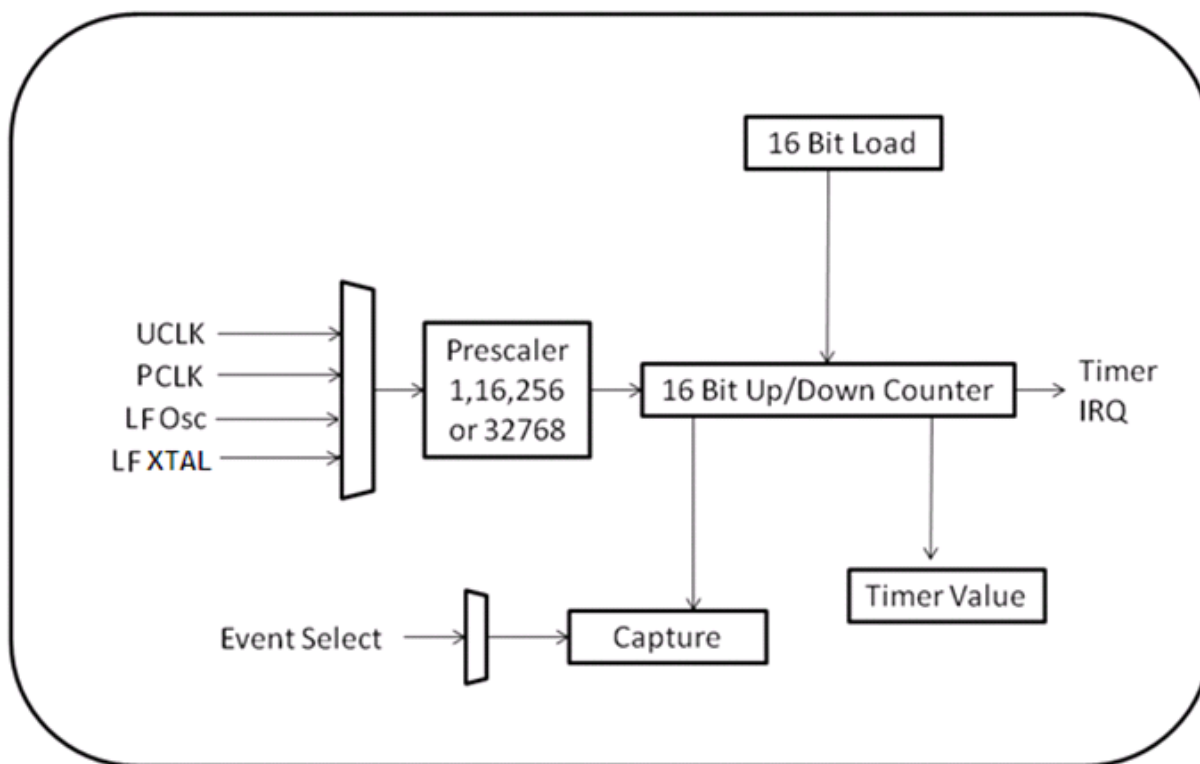


Figure 53: General Purpose Timers Block Diagram

## General Purpose Timers Operation

The general purpose timers have two modes of operation, free running or periodic.

In free running mode, the timer is started by writing TxCON enable bit. The timer increments from zero/full scale to full scale/zero if counting up/down. Full scale is  $2^{16}-1$  or 0xFFFF in binary format. On reaching full scale (or zero), a time out interrupt occurs and TxSTA[0] is set. To clear it, user code must write 1 to TxCLR[0]. If TxCON[7] is set, the timer will keep counting and reload when the TxCLR register is written.

In Periodic mode, the initial TxLD value should be loaded before enabling the timer. The timer is started by writing TxCON enable bit. The timer value increments from the value stored in the TxLD register to full scale or decrements from the value stored in the TxLD register to zero, depending on the TxCON[2] settings (count up/down). On reaching full scale or zero, the timer generates an interrupt. The TxLD is reloaded into the TxVAL and the timer continues counting up/down. The timer should be disabled prior to changing the TxCON or TxLD register. If the TxLD register is changed while the timer is timing out, undefined results can occur. By default the counter is reloaded automatically when generating the IRQ. If TxCON[7] is set to one, the counter is also reloaded when user code writes TxCLR. This allows user changes to the TxLD to take effect immediately and not on the next time out.

TxSTA should be read prior to writing to any timer registers following the set or clear of

enable. Once TxSTA[7] is cleared, registers can be modified. This assures the timer is done synchronizing timer control between the core and timer clock domains. The typical synchronization time is two timer clock periods.

At any time, TxVAL will contain a valid value to be read, synchronized to the APB clock. The TxCON register enables the counter, selects the mode, selects the prescale value, and controls the event capture function.

#### **Capture event function:**

There are 30 interrupt events that can be captured by the general purpose timers. These events are divided into two groups of 16 inputs for each of the timers as shown in Table 272. Any one of the 16 events associated with a general purpose timer can cause a capture of the 16 bit TxVAL register in to the 16 bit TxCAP register. TxCON has a 5-bit field selecting which of the 16 events to capture.

When the selected IRQ occurs, the TxVALS register is copied in to the TxCAP register. TxSTA[1] is set. The IRQ is cleared by writing 1 to bit[1] of the TxCLR1 register. The TxCAP register will also hold its value and cannot be overwritten until TxCLR1[1] is written with a 1.

**Table 272: Timer Capture Event**

Event select bits (TxCON[])	Timer 0 capture source	Timer1 capture source
0	Wake Up Timer.	Timer0
1	External interrupt 0	SPI0
2	External interrupt 1	SPI1
3	External interrupt 2	I2C slave
4	External interrupt 3	I2C master
5	External interrupt 4	Reserved
6	External interrupt 5	DMA error
7	External interrupt 6	DMA done – any of the DMA channels
8	External interrupt 7	External interrupt 1
9	UHF Transceiver Interrupt	External interrupt 2
10	Watchdog timer	External interrupt 3
11	Reserved	PWMTRIP
12	Timer1	PWMIRQ0
13	ADC	PWMIRQ1
14	Flash Controller	PWMIRQ2
15	UART	PWMIRQ3

# General Purpose Timers Memory Mapped Registers

## General Purpose Timer0 Register Map

**Table 273: General Purpose Timer0 Memory Mapped Registers Address Table**

Timer0 base address 0x40000000

Offset	Name	Description	Access	Default
0x0000	T0LD	16-bit load value	RW	0x0000
0x0004	T0VAL	16-bit timer value, read only.	R	0x0000
0x0008	T0CON	Control register	RW	0x000A
0x000C	T0CLR	Clear interrupt register	RW	0x0000
0x0010	T0CAP	Capture register	R	0x0000
0x001C	T0STA	Status register	R	0x0000

## General Purpose Timer1 Register Map

**Table 274: General Purpose Timer1 Memory Mapped Registers Address Table**

Timer1 base address 0x40000400

Offset	Name	Description	Access	Default
0x0000	T1LD	16-bit load value	RW	0x0000
0x0004	T1VAL	16-bit timer value, read only.	R	0x0000
0x0008	T1CON	Control register	RW	0x000A
0x000C	T1CLR	Clear interrupt register	RW	0x0000
0x0010	T1CAP	Capture register	R	0x0000
0x001C	T1STA	Status register	R	0x0000

## General Purpose Timers Load Registers

**Table 275: T0LD and T1LD Register Bit Description**

T0LD address 0x40000000 T1LD address 0x40000400

Bits	Name	Description
15 to 0	VALUE	Load value, 0 by default.

## General Purpose Timers Value Registers

**Table 276: T0VAL and T1VAL Register Bit Description**

T0VAL address 0x40000004 T1VAL address 0x40000404

Bits	Name	Description
15 to 0	VALUE	Current counter value, read only.

## General Purpose Timers Control Registers

**Table 277: T0CON and T1CON Register Bit Description**

T0CON address 0x40000008

T1CON address 0x40000408

Bits	Name	Description
15 to 13	Reserved	Reserved. These bits should be written 0.
12	EVENTEN	Event select bit. 1: Enable time capture of an event. 0: Cleared by user. (default)
11 to 8	EVENT	Event select range (0 to 15). The events are described in Table 272: Timer Capture .
7	RLD	Reload control bit, for periodic mode. 1: Change the load value on a write to TxCLRI. 0: reload only on a time out.
6 to 5	CLK	Clock select. 00: UCLK (default). 01: PCLK 10: LFOSC(32kHz internal oscillator.) 11: LFX TAL (External crystal (32 or 65 kHz).)
4	ENABLE	Timer enable bit. 1: Enable the timer. The timer starts counting from its initial value, 0 if count up mode or 0xFFFF if count down mode. 0: Disable the timer. Clearing this bit resets the timer, including the TxVAL register.
3	MOD	Timer Mode. 1: Operate in periodic mode (default). 0: Operate in free running mode.
2	UP	Count up. 1: Timer to count up. 0: Timer to count down (default).



T0CON and T1CON Register Bit Description continued

Bits	Name	Description
1 to 0	PRE	Prescaler. 00: source clock/1 01: source clock/16. 10: source clock/256 (default). 11: source clock/32768. If the selected clock source is UCLK, then this setting results in a prescaler of 4.

The TxCON registers should not be written if the corresponding TxSTA [6] or TxSTA [7] are set.

Note: The timer clock needs to be 4 times slower than PCLK (the system clock), therefore care needs to be taken when selecting the prescaler at different CD values.

## General Purpose Timers Clear Interrupt Registers

Table 278: T0CLRI and T1CLRI Register Bit Description

T0CLRI address 0x4000000C

T1CLRI address 0x4000040C

Bits	Name	Description
15 to 2	Reserved	Reserved. These bits should be written 0.
1	CAP	Clear captured event interrupt 1: Clear a capture event interrupt. This bit always reads 0.
0	TMOUT	Clear timeout interrupt 1: Clear a timeout interrupt. This bit always reads 0.

Note: Ensure that the register write has fully completed before returning from the interrupt handler. Use the Data Synchronization Barrier (DSB) instruction if necessary.

## General Purpose Timers Capture Registers

Table 279: T0CAP and T1CAP Register Bit Description

T0CAP address 0x40000010

T1CAP address 0x40000410

Bits	Name	Description
15 to 0	VALUE	16-bit captured value. Read only. TxCAP will hold its value until TxCLR[1] is set by user code. If the same event occurs again, TxCAP will not be overwritten.

## General Purpose Timers Status registers

Table 280: T0STA and T1STA Register Bit Description

T0STA address 0x4000001C

T1STA address 0x4000041C

Bits	Name	Description
15 to 8	Reserved	Reserved. These bits should be written 0.
7	PDOK	T0CLR/T1CLR synchronization. 1: Set automatically when the T0CLR/T1CLR value is being updated in the timer clock domain, indicating that the timer's configuration is not yet valid. 0: When the interrupt is cleared in the timer's clock domain.
6	BUSY	Timer Busy 1: Timer not ready to receive commands to TxCON. Previous change of the TxCON value has not been synchronised in the timer clock domain. 0: Timer ready to receive commands to TxCON. The previous change of TxCON has been synchronised in the timer clock domain.
5 to 2	Reserved	Reserved. These bits should be written 0.
1	CAP	Capture event pending 1: Capture event is pending 0: No capture event is pending
0	TMOUT	Time out event occurred 1: Time out event has occurred For count up mode this is when the counter reaches full scale. For counting down mode this is when the counter reaches 0. 0: No time out event has occurred

# Wake Up Timer

## Wake Up Timer Features

The timer2 block consists of a 32-bit counter clocked from one of four different sources:

1. System clock (PCLK)
2. External crystal (LFXTAL)
3. Internal oscillator
4. External clock applied on P0.5

The selected clock source can be scaled down using a prescaler of 1, 16, 256 or 32,768.

The wake-up timer will continue to run when PCLK clock is disabled independently of the clock source used. It can be used in free running or periodic mode. In free running mode, the timer counts from 0x00000000 to 0xFFFFFFFF and starts again from 0x00000000. In periodic mode, the timer counts from 0x00000000 to T2WUFD.

In addition, the wake up timer has four specific time fields to compare with the wake up counter: T2WUFA/B/C/D. All four wakeup compare points can generate interrupts. When in free running mode, T2WUA/B/C/D need to be reconfigured in software to generate a periodic interrupt.

## Wake Up Timer Block Diagram

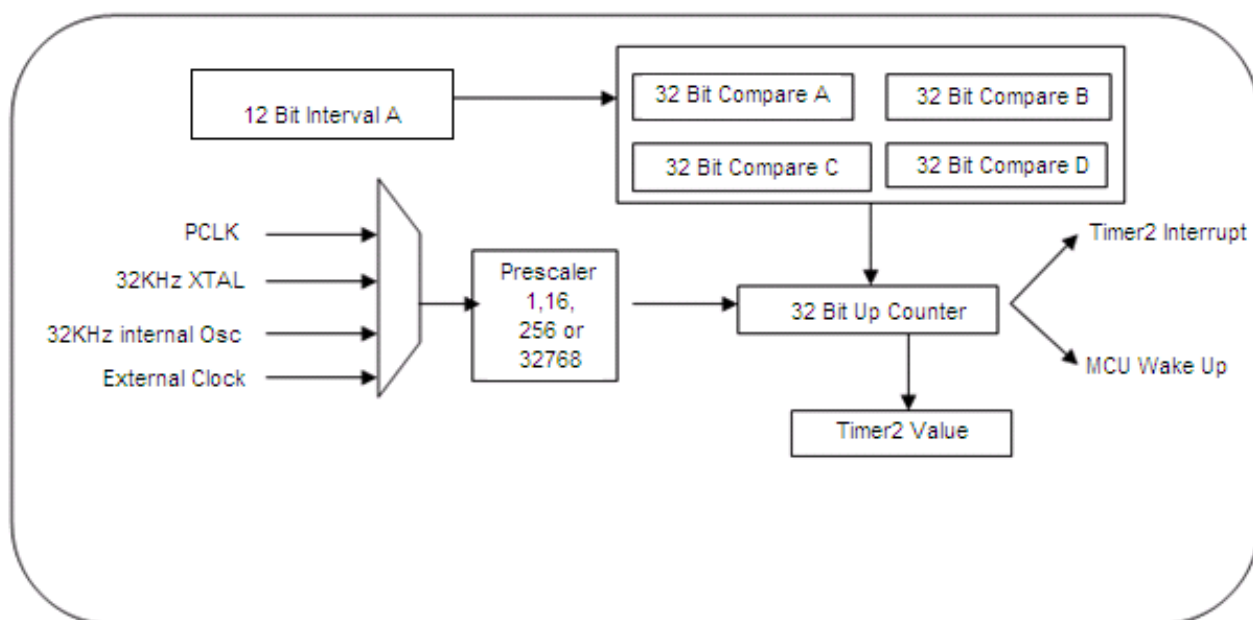


Figure 54: Wake up Timer Block Diagram

## Wake Up Timer Operation

The wake up timer comparator registers need to be configured before starting the timer. The timer is started by writing the control enable bit. The timer increments until the value reaches full scale in free running mode or when T2WUFD matches the wake up value T2VAL.

The T2VAL register can be read at any time, it reflects the current value of the counter. Since the Wake up timer is 32 bits but resides on a 16 bit bus, two bus reads are required to obtain the 32 bits. There are separate addresses for the upper (T2VAL1) and lower (T2VAL0) 16 bits of wake up timer. When the lower 16 bits is addressed and read, the upper 16 bits are latched and held in a separate register to be read later. The entire T2VAL register (upper and lower) remains frozen until the upper 16 bits are read. The control bit FREEZE must be set to freeze the T2VAL register between lower and upper reads.

### Interval register

T2INC is a 12-bit interval register. When a new value is written in T2INC, bits 16 to 5 of an internal 32-bit compare register (T2WUFA0/1) are loaded with the new T2INC value. This 32-bit compare register is automatically incremented with the contents of T2INC (shifted by 5) each time the wake up counter reaches the value in this compare register. This is providing that the new compare is less than T2WUFD value if in periodic mode or 0xFFFFFFFF if in free running mode. If the new compare value is greater than these limits, then it is recalculated as follow:

In free-running mode:  $\text{new T2WUFA} = \text{old T2WUFA} + \text{T2INC} - 0xFFFFFFFF$

In periodic mode:  $\text{new T2WUFA} = \text{old T2WUFA} + \text{T2INC} - \text{T2WUFD}$

The maximum programmable interval is just above 4s.

To modify the interval value, the timer must be stopped so that the interval register can be loaded in the compare register.

### Compare field registers

T2WUFB, T2WUFC and T2WUFD are 32-bit values programmed by the user in T2WUFx0 and T2WUFx1 registers (x = B, C or D). T2WUFD contains the load value when the wake up timer is configured in periodic mode.

In free running mode T2WUFB, T2WUFC and T2WUFD can be written to at any time but the corresponding interrupt enable in the T2IEN register must be disabled. Once the register is updated the interrupt can be re-enabled. In periodic mode this is only applicable to T2WUFB and T2WUFC.

### Interrupts/wake up signals

An IRQ is generated when the counter value correspond to any of the compare points or full scale in free running mode. The timer continues counting or is reset to 0.

The wake up timer generates 5 maskable interrupts. They are enabled in the T2IEN register. IRQ can be cleared by setting the corresponding bit in the T2CLR1 register.

Note that it takes two 32kHz clock cycles for the interrupt clear to take effect and during that time the ADuCRF101 should not be placed in any of the power down modes. IRQCRY indicates when the device can be placed in power down mode.

The timer is stopped and reset when clearing the timer enable bit in T2CON register.

## Wake Up Timer Memory Mapped Registers

**Table 281: Wake Up Timer Memory Mapped Registers Address Table**  
Base Address: 0x40002500

Offset	Name	Description	Access	Default
0x0000	T2VAL0	Current count value – LSB	R	0x0000
0x0004	T2VAL1	Current count value – MSB	R	0x0000
0x0008	T2CON	Control register	RW	0x0040
0x000C	T2INC	12-bit interval register for wake up field A	RW	0x00C8
0x0010	T2WUFB0	Wake up field B – LSB	RW	0x1FFF
0x0014	T2WUFB1	Wake up field B – MSB	RW	0x0000
0x0018	T2WUFC0	Wake up field C – LSB	RW	0x2FFF
0x001C	T2WUFC1	Wake up field C – MSB	RW	0x0000
0x0020	T2WUFD0	Wake up field D – LSB	RW	0x3FFF
0x0024	T2WUFD1	Wake up field D – MSB	RW	0x0000
0x0028	T2IEN	Interrupt enable	RW	0x0000
0x002C	T2STA	Status	R	0x0000
0x0030	T2CLRI	Clear interrupts.	W	N/A
0x003C	T2WUFA0	Wake up field A – LSB.	RW	0x1900
0x0040	T2WUFA1	Wake up field A – MSB.	RW	0x0000

### Wake up Timer Count Value Register

**Table 282: T2VAL Register Bit Description**

T2VAL0 Address: 0x40002500

T2VAL1 Address: 0x40002504

Bits	Name	Description
15 to 0	VALUE	Current Wake up timer Value

## Wake Up Timer Control Register

Table 283: T2CON Register Bit Description

Address: 0x40002508

Bits	Name	Description
15 to 12	Reserved	Unused bit locations
11	STOPINC	1: Stops the wakeup field A getting updated with the interval register value. This allows the user to update the interval register safely. 0: Allows the wakeup field A to be updated
10 to 9	CLK	Clock select: used to select clock source for timer: 00: PCLK (default) 01: 32K external oscillator 10: 32K internal oscillator 11: External clock, from P0.5
8	WUEN	Wake up enable bits for time field values: 1: Enable asynchronous wake up timer even when the core clock is off.( No APB clock is required) Once one of the time values equals the T2WUFA, the wake up output signal is generated. 0: Disable asynchronous wake up timer. Interrupt conditions will not wake up the part from sleep mode. This bit must be set to 1.
7	ENABLE	Timer Enable Bit. 1: Enable the timer 0: Disable the timer (default) This bit needs to be low when changing any of the control information or timer field values.
6	MOD	Timer Free Run Enable 1: Operate in free running mode (default). That is counts from 0 to 32'hFFFFFFF and starts again at 0. 0: Operate in periodic mode. That is it counts up to the value in T2WUFD
5 to 4	Reserved	Reserved
3	FREEZE	Freeze Enable Bit: 1: Enable the freeze of the high 16-bits after the lower bits have been read from T2VAL0. This ensures that the software will read an atomic shot of the timer. The entire T2VAL register unfreezes after the high bits (T2VAL1) have been read. 0: Disable this feature (default).
2	Reserved	Reserved
1 to 0	PRE	Clock Prescaler Select: 00: Source clock/1 (default) 01: Source clock/16. 10: Source clock/256. 11: Source clock/32,768. Note: If Clock prescaler select = 00 ( T2CON[1:0] = 00 ) and If the selected clock source is PCLK (T2CON[10:9] = 00) then this setting results in a prescaler of 4.

Note: The timer clock needs to be 4 times slower than PCLK (the system clock), therefore care needs to be taken when selecting the prescaler at different CD values.

## Wake Up Timer Interval Register: Interval Register

**Table 284: T2INC Register Bit Description**

Address: 0x4000250C

Bits	Name	Description
15 to 12	Reserved	Unused bit locations
11 to 0	VALUE	Wake up interval. This isn't used when format is non binary

T2INC is compared with bits [16:5] of the timer value. As it is shifted left by 5 bits its value needs to be multiplied by 32 to get the compare value.

With the default value of 0xC8:

For calculation purposes 0xC8 = 200 in decimal.

Interval =  $((200 * 32) + 1) \times 1/32768 = 195.3155\text{ms}$

(For a prescaler = 0 and 32K clock selected)

The value in T2INC can be updated while the time is enabled when STOPINC (T2CON [11]) in the control register is set. The new T2INC value takes effect after the next wake up field A interrupt. If the user is writing to this register while the timer is enabled the STOPINC bit should be set before writing to it and then clear STOPINC after the update.

## Wake up Timer Wake up Field Register B

T2WUFB0/1 registers can be written to at any time but the corresponding interrupt enable, T2IEN [1] must be disabled. Once the register is updated the interrupt can be re-enabled.

**Table 285: T2WUFB0 Register Bit Description**

Address: 0x40002510

Bits	Name	Description
15 to 0	VALUE	Lower 16 bits of Wake up field B

**Table 286: T2WUFB1 Register Bit Description**

Address: 0x40002514

Bits	Name	Description
15 to 0	VALUE	Upper 16 bits of Wake up field B

## Wake up Timer Wake up Field Register C

T2WUFC0/1 registers can be written to at any time but the corresponding interrupt enable, T2IEN [2] must be disabled. Once the register is updated the interrupt can be re-enabled.

**Table 287: T2WUFC0 Register Bit Description**

Address: 0x40002518

Bits	Name	Description
15 to 0	VALUE	Lower 16 bits of Wake up field C

**Table 288: T2WUFC1 Register Bit Description**

Address: 0x4000251C

Bits	Name	Description
15 to 0	VALUE	Upper 16 bits of Wake up field C

## Wake up Timer Wake up Field Register D

For periodic mode T2WUFD0/1 registers can be only be written to when the timer is disabled.

In free running mode T2WUFD0/1 registers can be written to while the timer is running.

Before doing so the corresponding interrupt enable, T2IEN [3] must be disabled. Once the register is updated the interrupt can be re-enabled.

**Table 289: T2WUFD0 Register Bit Description**

Address: 0x40002520

Bits	Name	Description
15 to 0	VALUE	Lower 16 bits of Wake up field D

**Table 290: T2WUFD1 Register Bit Description**

Address: 0x40002524

Bits	Name	Description
15 to 0	VALUE	Upper 16 bits of Wake up field D



## Wake Up Timer Interrupt Enable Register

**Table 291: T2IEN Register Bit Description**

Address: 0x40002528

Bits	Name	Description
15 to 5	Reserved	Unused bit locations
4	ROLL	Interrupt enable bit for instance when counter rolls over. Only occurs in free running mode. 1: Generate an interrupt when timer2 rolls over. 2: Disable the roll over interrupt (default).
3	WUFD	T2WUFD interrupt enable. 1: Generate an interrupt when T2VAL reaches T2WUFD. 0: Disable T2WUFD interrupt (default).
2	WUFC	T2WUFC interrupt enable. 1: Generate an interrupt when T2VAL reaches T2WUFC. 0: Disable T2WUFC interrupt (default).
1	WUFB	T2WUFB interrupt enable. 1: Generate an interrupt when T2VAL reaches T2WUFB. 0: Disable T2WUFB interrupt (default).
0	WUFA	T2WUFA interrupt enable. 1: Generate an interrupt when T2VAL reaches T2WUFA. 0: Disable T2WUFA interrupt (default).

## Wake Up Timer Status Register

Table 292: T2STA Register Bit Description

Address: 0x4000252C

Bits	Name	Description
15 to 9	Reserved	Unused bit locations
8	PDOK	Indicates when a change in the enable bit is synchronised to the 32K clock domain. It is set high when the Enable bit (bit 5) in the Control register is set or cleared. It returns low when the change in the Enable bit has been synchronised to the 32K clock domain.
7	FREEZE	Status of T2VAL freeze Set high when the T2VAL0 is read, indicating T2VAL is frozen Reset low when T2VAL1 is read, indicating T2VAL is unfrozen
6	IRQCRY	Status of wake up signal to power down control Set automatically when any of the interrupts are still set in the external crystal clock domain. Cleared automatically when the interrupts are cleared.
5	Reserved	Unused bit locations
4	ROLL	Interrupt status bit for instance when counter rolls over. Only occurs in free running mode. This bit goes high when enabled in the interrupt enable register and the T2VALS counter register is equal to all 1s.
3	WUFD	T2WUFD interrupt flag. 1: Indicate a comparator interrupt has occurred. 0: After a write to the corresponding bit in T2CLRI.
2	WUFC	T2WUFC interrupt flag. 1: Indicate a comparator interrupt has occurred. 0: After a write to the corresponding bit in T2CLRI.
1	WUFB	T2WUFB interrupt flag. 1: Indicate a comparator interrupt has occurred. 0: After a write to the corresponding bit in T2CLRI.
0	WUFA	T2WUFA interrupt flag. 1: Indicate a comparator interrupt has occurred. 0: After a write to the corresponding bit in T2CLRI.

## Wake Up Timer Clear Interrupt Register

**Table 293: T2CLRI Register Bit Description**

Address: 0x40002530

Bits	Name	Description
15 to 5	Reserved	Unused bit locations
4	ROLL	Interrupt clear bit for instance when counter rolls over. Only occurs in free running mode.
3	WUFD	T2WUFD interrupt flag. 1:Clear a T2WUFD interrupt flag Cleared automatically after synchronization.
2	WUFC	T2WUFC interrupt flag. 1:Clear a T2WUFC interrupt flag Cleared automatically after synchronization.
1	WUFB	T2WUFB interrupt flag. 1:Clear a T2WUFB interrupt flag Cleared automatically after synchronization.
0	WUFA	T2WUFA interrupt flag. 1:Clear a T2WUFA interrupt flag Cleared automatically after synchronization.

Note: Ensure that the register write has fully completed before returning from the interrupt handler. Use the Data Synchronization Barrier (DSB) instruction if necessary.

## Wake up Timer Compare Register A

T2WUFA0/1 registers can be written to while the timer is enabled.

T2WUFA0/1 is an asynchronous register in the 32K clock domain. Therefore it should be read twice to confirm the correct value has been read.

**Table 294: T2WUFA0 Register Bit Description**

Address: 0x4000253C

Bits	Name	Description
15 to 0	VALUE	Lower 16 bits of Compare Register A

**Table 295: T2WUFA1 Register Bit Description**

Address: 0x40002540

Bits	Name	Description
15 to 0	VALUE	Upper 16 bits of Compare Register A

# Watchdog Timer

## Watchdog Timer Features

The Watchdog timer is used to recover from an illegal software state. Once enabled by the user code, it requires periodic servicing to prevent it from forcing a reset or interrupt of the processor.

The Watchdog timer is clocked by the internal 32.768kHz oscillator. It is clocked at all times except during reset.

The Watchdog timer is a 16bit count-down timer with a programmable prescaler. The prescaler source is selectable which can be scaled by factors of 1, 16, 256, or 4096.

A Watchdog timer time-out can generate a reset or an IRQ. The T3CON[1] bit is added to allow selecting an IRQ instead of a reset, this can be used for debug. The IRQ can be cleared by writing 0xCCCC to the T3CLRI write-only register.

## Watchdog Timer Block Diagram

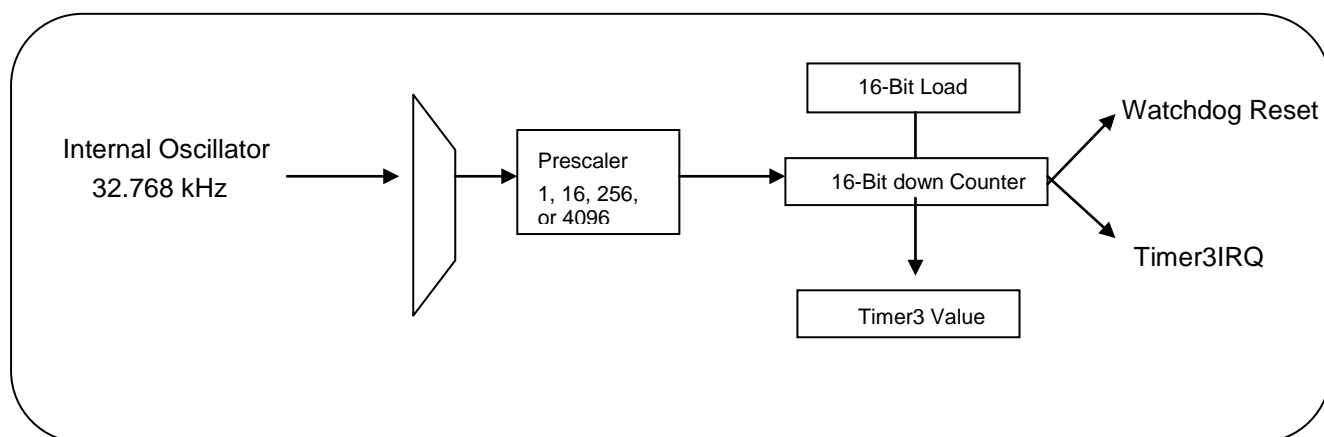


Figure 55: Watchdog Timer Block Diagram

## Watchdog Timer Operation

After a valid reset, the watchdog timer is initialized in hardware as follows.

```

T3CON = 0x00E9;
T3LD = 0x1000;
T3VAL = 0x1000;
  
```

This enables the Watchdog timer with a time out of 32s. This initial configuration can be modified by user code however setting T3CON [5] write protects T3CON and T3LD. This means that after kernel execution user code can disable the timer once, then reconfigure it with T3CON [5] set, once only. Only a reset will clear T3CON [5] and allow reconfiguring

the timer. If T3CON is not modified, user code can change T3LD at any time. If T3CON [5] is cleared, the timer is disabled. Settings can be modified and the timer re-enabled. If the Watchdog timer is set to free-running (T3CON[6] = 0), the Watchdog timer value will decrement from 0x1000 to zero, and wrap-around to 0x1000 and continue to decrement. To achieve a timeout value greater/less than 0x1000 (~32s with default prescale = 2), then periodic mode should be used (T3CON [6] = 1) and T3LD and T3CON [3:2] (prescale) written with the values corresponding to the desired timeout period. The maximum timeout is ~8192sec (T3LD=0xFFFF, prescale = 3). At any time, T3VAL will contain a valid value to be read, synchronized to the APB clock.

When the Watchdog timer decrements to 0 a reset (or IRQ) is generated. This reset can be prevented by writing T3CLRI with 0xCCCC before the expiration period. A write to T3CLRI will cause the Watchdog timer to reload with the T3LD (or 0x1000 if free-running mode) immediately to begin a new time-out period and start to count again. If any value other than 0xCCCC is written, a reset is generated (or IRQ if selected by T3CON [1]).

## Watchdog Timer Memory Mapped Registers

**Table 296: Watchdog Timer Memory Mapped Registers Address Table**

Base Address: 0x40002580

Offset	Name	Description	Access	Default
0x0000	T3LD	Load value.	RW	0x1000
0x0004	T3VAL	Current count value, read only.	R	0x1000
0x0008	T3CON	Control register.	RW	0x00E9
0x000C	T3CLRI	Clear interrupt.	W	N/A
0x0018	T3STA	Status register.	R	0x0020

### Watchdog Timer Load Register

**Table 297: T3LD Register Bit Description**

Address: 0x40002580

Bits	Name	Description
15 to 0	VALUE	Load value.

### Watchdog Timer Current Value Register

**Table 298: T3VAL Register Bit Description**

Address: 0x40002584

Bits	Name	Description
15 to 0	VALUE	Current count value. This register is synchronized to the APB clock. Read only register.

## Watchdog Timer Control Register

**Table 299: T3CON Register Bit Description**  
Address: 0x40002588

Bits	Name	Description
15 to 7	Reserved	Reserved. These bits should be written 0.
6	MOD	Timer Mode. 1: Operate in periodic mode (default). 0: Operate in free running mode. Note that in free running it wraps around at 0x1000.
5	ENABLE	Timer enable. 1: Enable the timer (default). 0: Disable the timer. Can only be done once.
4	Reserved	Reserved
3 to 2	PRE	Prescaler. 00: source clock/1. If the selected clock source is core clock, then this setting results in a prescaler of 4. 01: source clock/16. 10: source clock/256 (default). 11: source clock/4096.
1	IRQ	Timer interrupt. 1: Generate an interrupt when the timer times out. This feature is provided for debug purposes and is only available in ACTIVE mode. 0: Generate a reset on a time out (default).
0	PD	Power down clear. 1: reset, reload and restart the timer count upon exit from power down modes (default). 0: continue the count upon exit from power down modes.

### Summary of Events That Affect Timer Values:

1. Disabling timer: If T3CON [5] is reset on the initial T3CON write, the timer is reset and disabled.
2. Re-enabling timer: If T3CON [5] is set in a subsequent T3CON write, the T3 is reset, reloaded and restarted. Depending on the state of T3CON [6] (mode), T3 restarts from 0x1000 (free running mode, T3CON [6] = 0), or (periodic mode T3CON [6] = 1) gets the T3LD value and starts from there.
3. Write to T3CLRI to restart T3: The user will write to register T3CLRI with write data 0xCCCC within the timeout period to restart the T3 normally to prevent reset or IRQ.

## Watchdog Timer Clear Interrupt Register

**Table 300: T3CLRI Register Bit Description**

Address: 0x4000258C

Bits	Name	Description
15 to 0	VALUE	Clear watchdog. User writes 0xCCCC to reset/reload/restart T3 or clear IRQ. A write of any other value causes a watchdog reset/IRQ. Write only, reads 0.

Note: Ensure that the register write has fully completed before returning from the interrupt handler. Use the Data Synchronization Barrier (DSB) instruction if necessary.

## Watchdog Timer Status Register

The T3STA is a read only status register. Due to the asynchronous relationship and frequency difference between the APB clock and Timer3 clock, changes to the Watchdog timer configuration are synchronized between the two clock domains. Several of the status bits are used to signal the APB to T3 clock synchronization is in progress. The user can use these bits to verify the previous timer configuration write has taken effect in the T3 clock domain if necessary.

The status bits typically only need to be used if the Watchdog timer is disabled on the initial write to T3CON, followed by T3 configuration changes, followed by an immediate re-enable of T3. To be sure all changes are in effect; the re-enable should not be done until all “in progress” status bits are cleared.

**Table 301: T3STA Register Bit Description**

Address: 0x40002598

Bits	Name	Description
15 to 4	Reserved	Reserved.
3	CON	T3CON write sync in progress. 1: APB T3CON register values are being synchronized to T3 clock domain. 0: APB and T3 clock domains T3CON configuration values match.
2	LD	T3LD write sync in progress. 1: APB T3LD value is being synchronized to T3 clock domain. 0: APB and T3 clock domains T3LD values match.
1	CLRI	T3CLRI write sync in progress. 1: APB T3CLRI write is being synced to T3 clock domain. T3 will be restarted (if 0xCCCC was written) once sync is complete. 0: APB T3CLRI write sync not done/inactive.
0	IRQ	WD_IRQ 1: T3 interrupt is pending 0: T3 interrupt not pending

# PWM

## PWM Features

The ADuCRF101 integrates an eight channel PWM interface. Eight channels are grouped as 4 pairs (0-3). In addition to the standard mode on 4 channels, the first two pairs of PWM outputs (PWM0-1) can be configured to drive a H-bridge. On power up, the PWM outputs default to H-bridge mode. In the standard mode users have control over the period of each pair of outputs and over the duty cycle of each individual outputs. On power up, PWMCON0 defaults to 0X12 (HOFF = 1 and HMODE = 1). The PWM trip interrupt can be cleared by writing '1' to bit 4 of the PWMCLRI register. When using the PWM trip interrupt, the PWM interrupt should be cleared before exiting the ISR. This prevents the generation of multiple interrupts.

**Table 302: PWM Channel Grouping**

Port Name	Type	Description
PWM0	Output	High Side PWM output for pair0
PWM1	Output	Low Side PWM output for pair0
PWM2	Output	High Side PWM output for pair1
PWM3	Output	Low Side PWM output for pair1
PWM4	Output	High Side PWM output for pair2
PWM5	Output	Low Side PWM output for pair2
PWM6	Output	High Side PWM output for pair3
PWM7	Output	Low Side PWM output for pair3

## Operation in Standard Mode

In all modes the PWMxCOMx MMRs controls the point at which the PWM output changes state. The PWM clock is selectable via PWMCON0 with one of the following values: UCLK divided by 2, 4, 8, 16, 32, 64, 128 or 256.

The length of the PWM period is defined by PWMxLEN. Each pair has an associated counter.

The PWM waveforms are set by the count value of the 16 bit timer and the compare register contents.

The low-side waveform, PWM1, goes high when the timer count reaches PWM0LEN, and it goes low when the timer count reaches the value held in PWM0COM2 or when the high-side waveform PWM0 goes low.

The high-side waveform, PWM0, goes high when the timer count reaches the value held in PWM0COM0, and it goes low when the timer count reaches the value held in PWM0COM1.



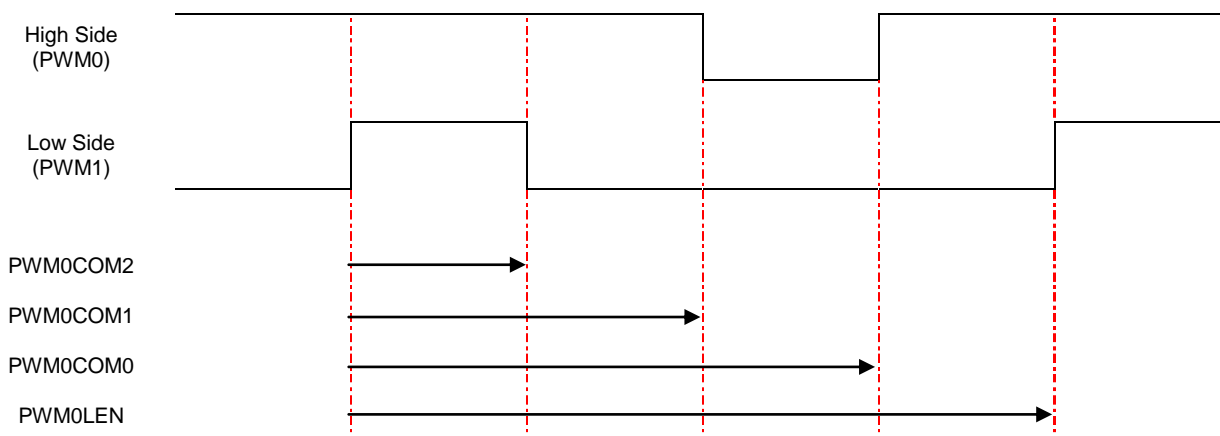


Figure 56: Waveform of PWM Channel Pair in Standard Mode

**Table 303: Compare Register Description**  
Base Address: 0x40001000

Name	Offset	Description
PWM0COM0	0X010	PWM0 output goes high when the PWM timer reaches the count value stored in this register
PWM0COM1	0X014	PWM0 output goes low when the PWM timer reaches the count value stored in this register
PWM0COM2	0X018	PWM1 output goes low when the PWM timer reaches the count value stored in this register
PWM0LEN	0X01C	PWM1 output goes high when the PWM timer reaches the count value stored in this register
PWM1COM0	0X020	PWM2 output goes high when the PWM timer reaches the count value stored in this register
PWM1COM1	0X024	PWM2 output goes low when the PWM timer reaches the count value stored in this register
PWM1COM2	0X028	PWM3 output goes low when the PWM timer reaches the count value stored in this register
PWM1LEN	0X02C	PWM3 output goes high when the PWM timer reaches the count value stored in this register
PWM2COM0	0X030	PWM4 output goes high when the PWM timer reaches the count value stored in this register
PWM2COM1	0X034	PWM4 output goes low when the PWM timer reaches the count value stored in this register
PWM2COM2	0X038	PWM5 output goes low when the PWM timer reaches the count value stored in this register
PWM2LEN	0X03C	PWM5 output goes high when the PWM timer reaches the count value stored in this register
PWM3COM0	0X040	PWM6 output goes high when the PWM timer reaches the count value stored in this register
PWM3COM1	0X044	PWM6 output goes low when the PWM timer reaches the count value stored in this register
PWM3COM2	0X048	PWM7 output goes low when the PWM timer reaches the count value stored in this register
PWM3LEN	0X04C	PWM7 output goes high when the PWM timer reaches the count value stored in this register

Table 304 below lists equations for the period and low duration for both the outputs of a PWM channel.

**Table 304: PWM Equations**

	Period	
Low Side (PWM1)	$T_{\text{uclk}} \times (\text{PWM0LEN} + 1) \times N_{\text{prescale}}$	High Duration : If (PWMCOM2 < PWMCOM1) then $T_{\text{uclk}} \times (\text{PWM0COM2} + 1) \times N_{\text{pre-scale}}$ Else $T_{\text{uclk}} \times (\text{PWM0COM1} + 1) \times N_{\text{pre-scale}}$
High Side (PWM0)	$T_{\text{uclk}} \times (\text{PWM0LEN} + 1) \times N_{\text{prescale}}$	Low Duration : $T_{\text{uclk}} \times (\text{PWM0COM0} - \text{PWM0COM1}) \times N_{\text{pre-scale}}$

Note:  $T_{\text{uclk}}$  = Period of UCLK clock input  $N_{\text{prescale}}$  = Prescalar value as determined by PWMCON0 [8:6]

## PWM Interrupt Generation

### PWM Tripn Enable Pin:

When the PWM Tripn pin is enabled (PWMCON1 [6]) and the Tripn signal goes low (falling edge), the PWM disables itself. It will also generate the PWM trip interrupt. The interrupt is cleared by setting PWMCLRI [4].

### PWM Interrupts:

The PWM has 4 interrupts: IRQPWM0, IRQPWM1, IRQPWM2 and IRQPWM3.

When the IRQ generation is enabled (PWMCON0 [10]) and the counter value for Pair-0 changes from PLW0LEN to 0 it will also generate the IRQPWM0 interrupt.

The interrupt is cleared by setting PWMCLRI [0].

When the IRQ generation is enabled (PWMCON0 [10]) and the counter value for Pair-1 changes from PLW1LEN to 0 it will also generate the IRQPWM1 interrupt.

The interrupt is cleared by setting PWMCLRI [1].

When the IRQ generation is enabled (PWMCON0 [10]) and the counter value for Pair-2 changes from PLW2LEN to 0 it will also generate the IRQPWM2 interrupt.

The interrupt is cleared by setting PWMCLRI [2].

When the IRQ generation is enabled (PWMCON0 [10]) and the counter value for Pair-3 changes from PLW3LEN to 0 it will also generate the IRQPWM3 interrupt.

The interrupt is cleared by setting PWMCLRI [3].

## PWM Memory Mapped Registers

**Table 305: PWM Memory Mapped Registers Address Table**  
**Base Address: 0x40001000**

Offset	Name	Description	Access	Default
0x000	PWMCON0	PWM Control register 0	RW	0x0012
0x004	PWMCON1	Trip control register	RW <sup>1</sup>	0x0000
0x008	PWMCLRI	PWM interrupt clear.	W	0x0000
0x010	PWM0COM0	Compare Register 0 for PWM0 and PWM1	RW	0x0000
0x014	PWM0COM1	Compare Register 1 for PWM0 and PWM1	RW	0x0000
0x018	PWM0COM2	Compare Register 2 for PWM0 and PWM1	RW	0x0000
0x01C	PWM0LEN	Period Value register for PWM0 and PWM1	RW	0x0000
0x020	PWM1COM0	Compare Register 0 for PWM2 and PWM3	RW	0x0000
0x024	PWM1COM1	Compare Register 1 for PWM2 and PWM3	RW	0x0000
0x028	PWM1COM2	Compare Register 2 for PWM2 and PWM3	RW	0x0000
0x02C	PWM1LEN	Period Value register for PWM2 and PWM3	RW	0x0000
0x030	PWM2COM0	Compare Register 0 for PWM4 and PWM5	RW	0x0000
0x034	PWM2COM1	Compare Register 1 for PWM4 and PWM5	RW	0x0000
0x038	PWM2COM2	Compare Register 2 for PWM4 and PWM5	RW	0x0000
0x03C	PWM2LEN	Period Value register for PWM4 and PWM5	RW	0x0000
0x040	PWM3COM0	Compare Register 0 for PWM6 and PWM7	RW	0x0000
0x044	PWM3COM1	Compare Register 1 for PWM6 and PWM7	RW	0x0000
0x048	PWM3COM2	Compare Register 2 for PWM6 and PWM7	RW	0x0000
0x04C	PWM3LEN	Period Value register for PWM6 and PWM7	RW	0x0000

---

<sup>1</sup> RW8 is 8 bits read or write.

## PWM Control Register 0

**Table 306: PWMCON0 Register Bit Description**  
**Address:0x40001000**

Bit Position	Name	Description
15	SYNC	PWM Synchronization. 1: All PWM counters are reset on the next clock edge after the detection of a falling edge on the SYNC pin. 0: Ignore transitions on the SYNC pin.
14	PWM7INV	1: Invert PWM7. 0: PWM7 is normal
13	PWM5INV	1: Invert PWM5. 0: PWM5 is normal
12	PWM3INV	1: Invert PWM3. 0: PWM3 is normal
11	PWM1INV	1: Invert PWM1. 0: PWM1 is normal
10	PWMIEN	1: Enables PWM interrupts. 0: disables the PWM interrupts.
9	ENA	If HOFF = 0 and HMODE = 1.( If not in H-Bridge mode, this bit has no effect.) 1: Enable PWM outputs. 0: Disable PWM outputs.
8 to 6	PWMCP	PWM Clock Prescaler. Sets UCLK divider. 000: UCLK/2. 001: UCLK/4. 010: UCLK/8. 011: UCLK/16. 100: UCLK/32. 101: UCLK/64. 110: UCLK/128. 111: UCLK/256.
5	POINV	1: Invert all PWM outputs. 0: PWM outputs as normal.
4	HOFF	High Side Off. 1: Forces PWM0 and PWM2 outputs high and PWM1 and PWM3 low.(Default) 0: PWM outputs as normal.
3	LCOMP	Load Compare Registers. 1: Load the internal compare registers with the values in PWMxCOMx on the next transition of the PWM timer from 0x00 to 0x01. 0: Use the values previously stored in the internal compare registers.
2	DIR	Direction Control. 1: Enables PWM0 and PWM1 as the output signals while PWM2 and PWM3 are held low. 0: Enable PWM2 and PWM3 as the output signals while PWM0 and PWM1 are held low.

**PWMCON0 Register Bit Description continued**

Bit Position	Name	Description
1	HMODE	1: Enables H-Bridge mode and Bit 5:2 of PWMCON0. (Default) 0: PWMs in standard mode.
0	PWMEN	1: Enables all PWM outputs. 0: Disables all PWM outputs.

Note:

1. Except for LCOMP, all other bits of PWMCON0 register can be changed only when PWMEN is LOW.
2. When LCOMP is written with value '1', it will stay at that value until the new value has been loaded in compare registers for all the channels.

**PWM1 Control Register****Table 307: PWMCON1 Register Bit Description**

Address:0x40001004

Bit Position	Name	Description
15 to 7	Reserved	Reserved. Reads 0.
6	TRIPEN	1: Enable PWM trip functionality 0: Disable PWM trip functionality
5 to 0	Reserved	Reserved

**PWMCLRI Register****Table 308: PWMCLRI Register Bit Description**

Address:0x40001008

Bit Position	Name	Description
15 to 5	Reserved	Reserved. These bits always reads 0
4	TRIP	1: clear the latched TripIrqPwm interrupt This bit always reads 0
3	PWM3	1: clear the latched irqPwm3 interrupt This bit always reads 0
2	PWM2	1: clear the latched irqPwm2 interrupt This bit always reads 0

## PWMCLRI Register Bit Description continued

Bit Position	Name	Description
1	PWM1	1: clear the latched irqPwm1 interrupt This bit always reads 0
0	PWM0	1: clear the latched irqPwm0 interrupt This bit always reads 0

## Description of H-bridge Mode

For H-bridge mode HMODE =1 (PWMCON0 [1] = 1). The HMODE bit also works with PWMCON0 [5:2] for H bridge mode. Please note that only PWM0-3 participates in H-bridge mode, other outputs (PWM4-7) don't and continue to generate standard mode output. The HMODE bit also works with PWMCON0 [5:2] for H bridge mode.

**Table 309: PWM Output in H-bridge Mode**

PWM Control Bits				PWM OUTPUTS			
ENA PWMCON0[9]	POINV PWMCON0[5]	HOFF PWMCON0[4]	DIR PWMCON0[2]	PWM 0	PWM 1	PWM 2	PWM 3
0	X	0	X	1	1	1	1
X	X	1	X	1	0	1	0
1	0	0	0	0	0	HS	LS
1	0	0	1	HS	LS	0	0
1	1	0	0	~HS	~LS	1	1
1	1	0	1	1	1	~HS	~LS

# ADC Circuit Overview

## ADC Circuit Features

The analog-to-digital converter (ADC) peripheral provides the user with a multi-channel multiplexer, a 1.25V on-chip reference and a 12-bit ADC, capable of providing a throughput of up to 1MSPS.

The converter accepts an analog input range of 0V to LVDD or 0V to VREF depending on the reference selected. The absolute maximum rating on the ADC input is 2V.

Single or continuous conversion can be initiated in software. Timer0 or Timer1 overflow can also be used to trigger an ADC conversion.

Measurements on the external sensors can be ratio metric as shown in

Figure 57. The multiplexer allows selection of the following:

1. Up to 6 external ADC channels
2. Power supply monitoring
3. Internal temperature sensor

Digital gain and offset trim are also available in software.

## ADC Top Block Level Diagram

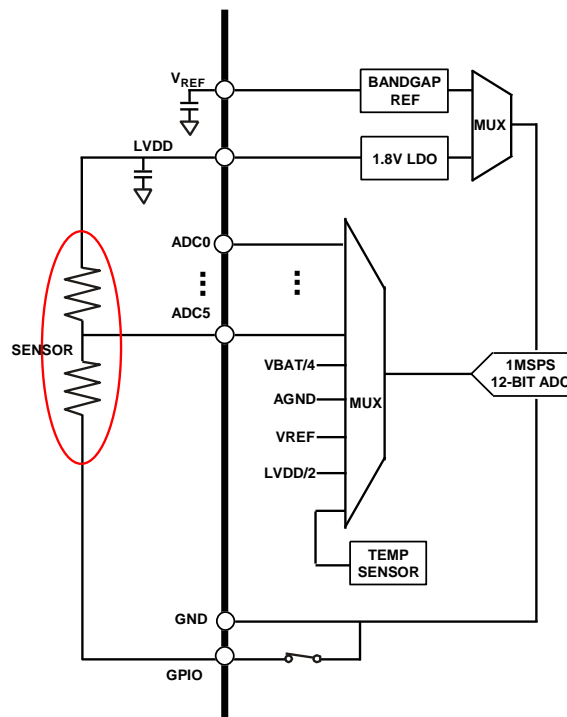


Figure 57: ADC Block Diagram

## ADC Input Structure

Figure 58 shows the equivalent circuit of the analog input structure of the ADC. The two diodes provide ESD protection for the analog inputs. Care must be taken that the analog input signals never exceed the supply rails by more than 300 mV.

The C1 capacitor is typically 4pF and can be primarily attributed to pin capacitance. The resistor is typically 750Ω and is made up of the on resistance of the switches mainly. The C2 capacitor is the ADC's sampling capacitor on the single ended input and is typically 16pF.

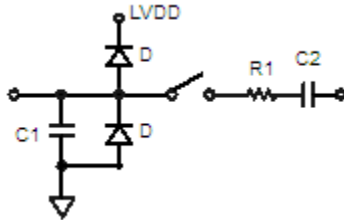


Figure 58: Equivalent Analog Input Circuit.

In acquisition phase, the switch is closed.

In conversion phase the switch is open.

## ADC Timing

Figure 59 gives details of the ADC timing. The ADC clock speed and acquisition clock are programmable. By default the acquisition time is 8 ADC clocks and the ADC clock is 8MHz. The number of extra clocks, for bit trial and register write is set to 12 + 2 which gives a default sampling frequency of 364 kHz.

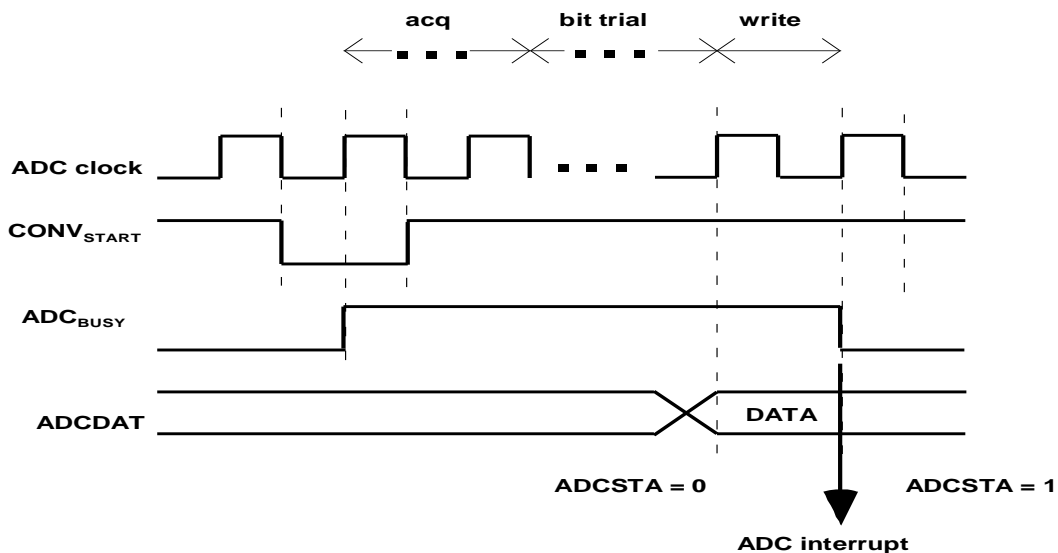


Figure 59: ADC Timing Internal Signals



## Bandgap Reference

The internal bandgap reference is provided to measure the VBAT/4 ADC input accurately. The internal 1.25V reference is not intended to drive any external circuitry. The reference needs an external 0.47 $\mu$ F capacitor.

The ADC input range is limited by the ADC reference. The LVDD channel accommodates measuring the accuracy of the LDO output versus the reference. This measurement is used in software compensations.

An external reference can be connected to the VREF pin and overdrive the internal reference when it is turned off. The maximum voltage for an external reference is 1.8V.

## Temperature Sensor

An internal temperature sensor is provided on chip to indicate the die temperature. The temperature sensor is not calibrated. Its accuracy is approximately  $\pm 10^{\circ}\text{C}$ . The temperature sensor buffer is turn on/off automatically when selecting the temperature sensor channel.

## VBAT and LVDD Attenuators

The ADC can measure two attenuated supplies, VBAT and LVDD. The attenuators are powered up/down automatically when selecting the corresponding channel in the ADCCFG MMR. The resistor size for these attenuators is of the order of 384k $\Omega$  and 128k $\Omega$  for VBAT and 256k $\Omega$  for LVDD.

## ADC Initialization

To ensure the validity of the first ADC sample, the following steps must be performed:

- Enable the reference buffer and ADC circuitry by writing ADCCON = 0x00.
- Configure the ADC channel for LVDD/2 with ADCCFG = 0x0A0C.
- Perform a software conversion (ADCCON = 0x01) until ADCDAT is close to the expected result or less than 3000.
- Reconfigure ADCCFG to the wanted channel and speed.
- Start ADC conversion by writing ADCCON, the reference should be settled.

This procedure optimises the length of the software delay required to allow the reference buffer to settle.

If the LDO is used as ADC reference, the similar initialisation is necessary.

Once configured via the ADC configuration and control registers, the ADC converts the analog input and provides a 12-bit result in the ADC data register. The status bit is set when the result is available.

## ADC Interrupt Generation

The status bit is set when new data is available and it will stay set in between concurrent new data if the data register is not read. Only a read of the data register, a power down of the ADC or a reset of the device clears the status bit.

If enabled in the NVIC and in the ADC control register, an interrupt will occur each time there is new data, regardless of whether the status bit is set or not.

## ADC Calibration

By default, the factory-set values written to the ADC offset (ADCOF) and gain coefficient registers (ADCGN) yield optimum performance in terms of end-point errors and linearity for standalone operation of the part. If system calibration is required, it is possible to modify the default offset and gain coefficients to improve end-point errors, but note that any modification to the factory-set ADCOF and ADCGN values can degrade ADC linearity performance.

**The offset and gain register does not affect the result in the ADCDAT register. The values in the offset and gain register must be read by the user and taken into account in software to yield optimum performance.**

For system offset error correction, the ADC channel input stage must be tied to AGND. A continuous software ADC conversion loop must be implemented by modifying the value in ADCOF until the ADC result reads code 0 to 1. If the ADC result value is greater than 1, ADCOF should be decremented until ADC result reads 0 to 1. Offset error correction is done digitally and has a resolution of 0.25 LSB and a range of  $\pm 3.125\%$  of the ADC reference.

For system gain error correction, the ADC channel input stage must be tied to VREF. A continuous software ADC conversion loop must be implemented to modify the value in ADCGN until the ADC result reads code 4094 to 4095. If the ADC result is less than 4094, ADCGN should be incremented until the ADC result reads 4094 to 4095. Similar to the offset calibration, the gain calibration resolution is 0.25 LSB with a range of  $\pm 3\%$  of the ADC reference.

# ADC Memory Mapped Registers

**Table 310: ADC Memory Mapped Registers Address Table**

Base Address: 0x40050000

Offset	Name	Description	Access	Default
0x00	ADCCFG	Configuration register	RW	0x0A00
0x04	ADCCON	Control register	RW	0x0090
0x08	ADCSTA	Status register	R	0x0000
0x0C	ADCDAT	12-bit data register	R	0x000
0x10	ADCGN	16-bit gain register	RW	Modified by the Kernel
0x14	ADCOF	16-bit offset register	RW	Modified by the Kernel

## ADC Configuration Register

**Table 311: ADCCFG Register Bit Description**  
**Address: 0x40050000**

Bits	Name	Description
15 to 14	Reserved	Reserved. These bits should be written 0.
13	LVDDREF	Reference select 1: Select LVDD as ADC reference. 0: Select the internal 1.25V reference as ADC reference (default).
12 to 10	CLK	ADC clock frequency 000: $f_{CORE}$ 001: $f_{CORE} / 2$ 010: $f_{CORE} / 4$ (default) 011: $f_{CORE} / 8$ 100: $f_{CORE} / 16$ 101: $f_{CORE} / 32$ Other: $f_{CORE} / 2$
9 to 8	ACQ	Acquisition clocks 00: 2 01: 4 10: 8 (default) 11: 16
7 to 4	Reserved	Reserved. These bits should be written 0.
3 to 0	CHSEL	Channel select: 0000: ADC0 (default) 0001: ADC1 0010: ADC2 0011: ADC3 0100: ADC4 0101: ADC5 0110: VBAT/4, supply monitor 0111: Reserved 1000: Reserved 1001: temperature sensor 1010: AGND – for offset calibration 1011: VREF – for gain calibration 1100: LVDD/2 supply monitor Other: ADC0

## ADC Control Register

Table 312: ADCCON Register Bit Description

Address: 0x4050004

Bits	Name	Description
15 to 8	Reserved	
7	REF	Internal reference buffer enable. 1: Turn off the reference buffer (default). The internal reference buffer must be turned off if using an external reference. 0: Turn on the reference buffer. The reference buffer takes 5ms to settle and consumes approximately 210uA.
6	Reserved	Reserved. This bit should be written to 0.
5	IRQ	IRQ enable 1: Enable the ADC interrupt. An interrupt is generated when new data is available. 0: Disable the ADC interrupt. (default)
4	PD	Power down mode 1: Power down the ADC. (default) 0: Power up the ADC.
3 to 1	MOD	Conversion modes: 000: software convert start (default) 001: continuous convert mode 010: Reserved 011: timer0 overflow 100: timer1 overflow 101: GPIO, P1.6 Other: software convert start.
0	START	ADC conversion start 1: Start conversion as per conversion mode. 0: Stop converting. This bit does not clear after a single software conversion (conversion mode 0) but does not need to be cleared by software before starting a new conversion.

## ADC Status Register

**Table 313: ADCSTA Register Bit Description**

Address: 0x4050008

Bits	Name	Description
15 to 1	Reserved	Reserved. These bits should be written 0.
0	READY	ADC ready bit 1: Set by the ADC when a conversion is complete and generates an interrupt if enabled. It is cleared automatically when ADCDAT is read.

## ADC Data Register

**Table 314: ADCDAT Register Bit Description**

Address: 0x405000C

Bits	Name	Description
15 to 14	Reserved	Reserved. These bits should read 0.
13 to 2	VALUE	12-bit ADC result. Reading this register clears the status bit. The ADC results stays in the register until a new value is available.
1 to 0	Reserved	Reserved. These bits should read 0.

## ADC Gain Register

**Table 315: ADCGN Register Bit Description**

Address: 0x4050010

Bits	Name	Description
15 to 0	VALUE	16-bit gain correction register

## ADC Offset Register

**Table 316: ADCOF Register Bit Description**

Address: 0x4050014

Bits	Name	Description
15 to 0	VALUE	16-bit offset correction register

# Power Supply Support Circuits

## Power Supply Support Circuits Features

### Low Drop Out Regulators

The ADuCRF101 integrates two on-chip regulators (LDO) that are driven directly from the battery voltage to generate 1.725V to 1.8V internal supply. This regulated supply is then used as supply voltage for the ARM Cortex-M3 and peripherals including the precision analog circuits on-chip. The two LDOs share the same output pin, LVDD. They need an external capacitor of 0.47uF for stability. One of the LDO (HP LDO) is used in active mode and can supply up to 20mA while the other LDO (LP LDO) can only supply 50uA maximum and is used for low power modes. Turning on and off the LDOs is done automatically when selecting the operation mode of the device. It is not possible to overdrive the LDO. The HP LDO can supply the external sensors for ratio metric measurements (5mA supply).

### Power-On-Reset

Power-on-reset (POR) function is also integrated to ensure safe operation of the MCU. The POR circuit is designed to guaranty full functional operation of the Flash/EE memory based MCU during power-on and power-down cycles.

As shown in Figure 60 when the supply voltage on VBAT reaches a minimum operating voltage of 1.6V, and the LDO output 1.65V, a POR signal keeps the core in reset for 32ms. The output of the POR is available on P1.1.

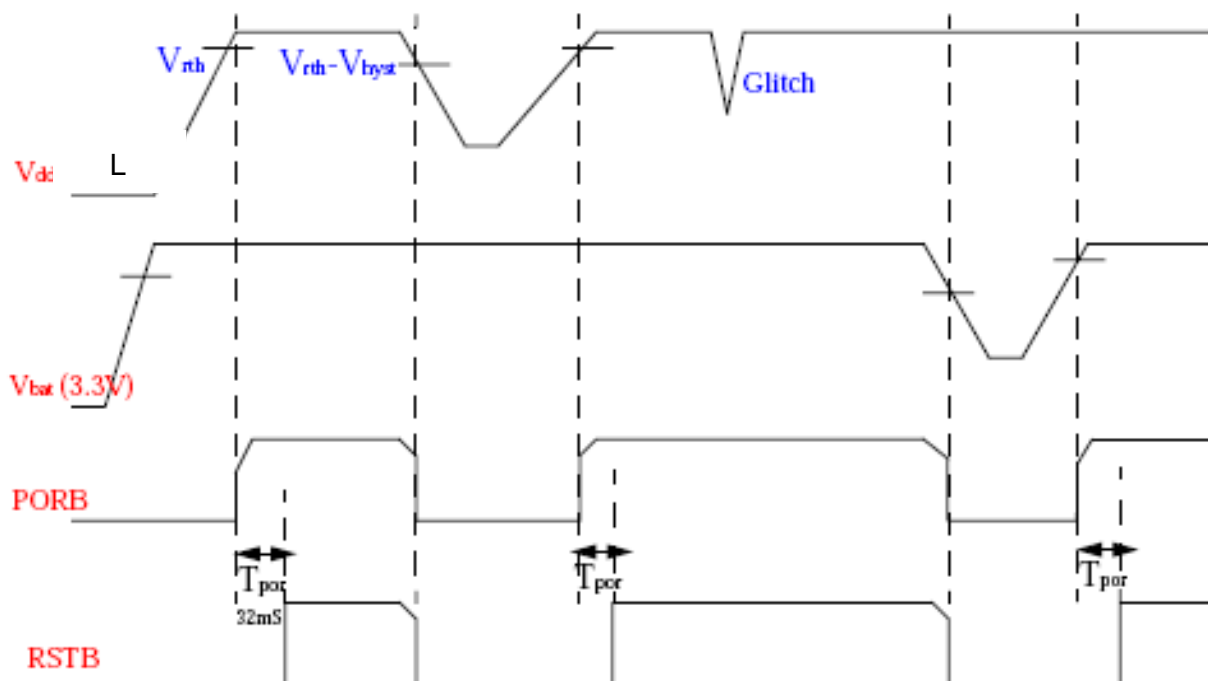


Figure 60: Typical Power-on Cycle.

## Power Supply Monitor

A Power Supply Monitor (PSM) is also available on-chip. It generates a non maskable interrupt if LVDD drops below 1.71V approximately.

The PSM can be disabled by user code. During low power mode the PSM must be disabled.

## Power Supply Monitor Memory Mapped Registers

**Table 317: PSM Memory Mapped Registers Address Table**

Base Address: 0x40002400

Offset	Name	Description	Access	Default
0x08	PSMCON	Configuration register	RW	0x1

### PSM Control Register

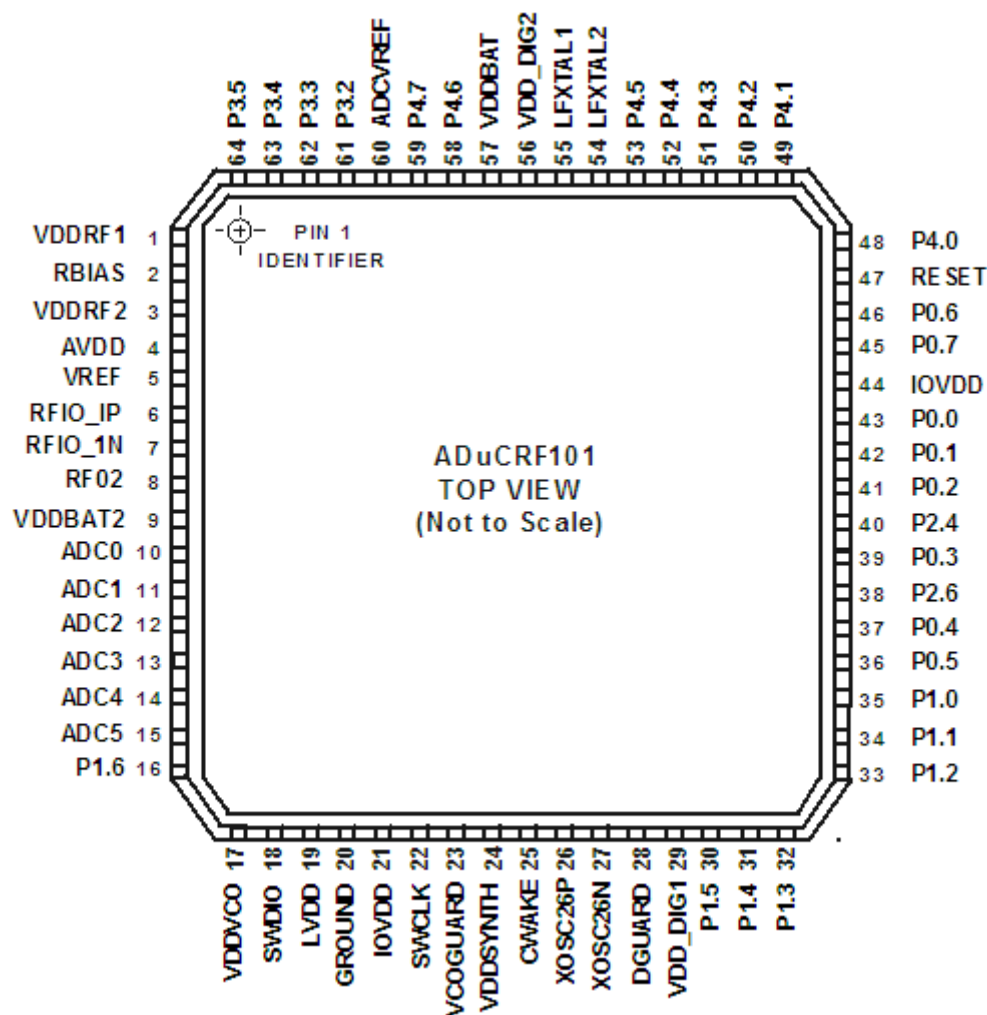
**Table 318: PSMCON Register Bit Description**

Address: 0x40002408

Bits	Name	Description
7 to 1	Reserved	Reserved. These bits should be written 0.
0	PD	Power down mode 1: Power down the PSM (default). 0: Power up the PSM.



## ADuCRF101 Pin Configuration



### Figure 61: ADUCRF101 Pin Configuration

**Table 319: ADuCRF101 Pin Function Descriptions**

Pin No.	Pin Name	Description
1	VDDRF1	Voltage Regulator output for RF block. A 220 nF capacitor should be placed between this pin and ground for regulator stability and noise rejection.
2	RBIAS	External bias resistor. Optimum resistor is 36 kΩ with 5% tolerance.
3	VDDRF2	Voltage Regulator output for RF block. A 220 nF capacitor should be placed between this pin and ground for regulator stability and noise rejection.
4	AVDD	Battery terminal, Analog Supply for ADC and ADC internal reference (VREF).
5	VREF	Internal 1.25V ADC reference. A 0.47μF capacitor between this pin and ground is required.
6	RFIO_1P	LNA Positive Input in Receive Mode. Differential PA Positive Output in Transmit Mode.
7	RFIO_1N	LNA Negative Input in Receive Mode. Differential PA Negative Output in Transmit Mode.
8	RF02	Single ended PA output.
9	VDD_VBAT2	Battery Terminal.
10	ADC0	ADC input channel 0
11	ADC1	ADC input channel 1
12	ADC2	ADC input channel 2
13	ADC3	ADC input channel 3
14	ADC4	ADC input channel 4
15	ADC5	ADC input channel 5
16	P1.6/IRQ7/ ADC_CONV_ST PWM_SYNC	General Purpose Input and Output Port 1.6/External Interrupt 7/ADC conversion start/PWM synchronisation.
17	VDDVCO	Voltage Regulator output for VCO. A 220 nF capacitor should be placed between this pin and ground for regulator stability and noise rejection.
18	SWDIO	Serial Wire debug data in-out
19	LVDD	On chip LDO decoupling output. Connect a 0.47μF capacitor to the 1.8V output to ensure core operating voltage is stable
20	GROUND	Ground. The exposed package paddle should be grounded.
21	IOVDD	General Purpose IO supply. Connect to the battery terminal
22	SWCLK	Serial Wire debug clock
23	VCOGUARD	Guard, screen for VCO.
24	VDDSYNTH	Voltage Regulator output for Synthesizer. A 220 nF capacitor should be placed between this pin and ground for regulator stability and noise rejection.
25	CWAKE	External capacitor for wake up control. A 150 nF capacitor should be placed between this pin and ground.
26	XOSC26P	The 26MHz reference crystal should be connected between this pin and XOSC26N.
27	XOSC26N	The 26MHz reference crystal should be connected between this pin and XOSC26P.
28	DGUARD	Internal Guard, Screen for Digital Cells

## ADuCRF101 Pin Function Descriptions continued

Pin No.	Pin Name	Description
29	VDD_DIG 1	Voltage Regulator output for Digital. A 220 nF capacitor should be placed between this pin and ground for regulator stability and noise rejection.
30	P1.5/IRQ6/ I2C0SDA/PWM7	General Purpose Input and Output Port 1.5/External Interrupt 6/I2C 0 Serial Data/PWM channel 7.
31	P1.4/IRQ5/ I2C0SCL/PWM6	General Purpose Input and Output Port 1.4/External Interrupt 5/I2C 0 Serial Clock/PWM channel 6.
32	P1.3/ PWM5	General Purpose Input and Output Port 1.3/ PWM channel 5.
33	P1.2/ PWM4	General Purpose Input and Output Port 1.2/ PWM channel 4.
34	P1.1/PORB/ TXD/ PWM3	General Purpose Input and Output Port 1.1/POR output/ UART TXD/PWM channel 3.
35	P1.0/ IRQ4/ RXD/PWM2	General Purpose Input and Output Port 1.0/ External Interrupt 4/ UART RXD/ PWM channel 2.
36	P0.5/CS2/ECLKIN	General Purpose Input and Output Port 0.5/SPI Chip Select 2/External Clock Input
37	P0.4/IRQ0/CS1 /ECLKOUT	General Purpose Input and Output Port 0.4/External Interrupt 0/SPI Chip Select 1/External Clock Output.
38	P2.6	General Purpose Input and Output Port 2.6
39	P0.3/IRQ1/CS0/ PWM1	General Purpose Input and Output Port 0.3/External Interrupt 1/SPI Chip Select 0/PWM channel 1.
40	P2.4/IRQ8	General Purpose Input and Output Port 2.4/ External Interrupt 8
41	P0.2/MOSI/PWM0	General Purpose Input and Output Port 0.2/SPI Master Out Slave In Pin (MOSI)/PWM channel 0.
42	P0.1/SCLK/	General Purpose Input and Output Port 0.1/SPI Serial Clock
43	P0.0/MISO/	General Purpose Input and Output Port 0.0/SPI Master In Slave Out Pin (MISO)
44	IOVDD	General Purpose I/O supply. Connect to the battery terminal
45	P0.7/IRQ3/CS4 /CTS	General Purpose Input and Output Port 0.7/ External Interrupt 3 / SPI Chip Select 4/ UART CTS
46	P0.6/ IRQ2/CS3/ RTS	General Purpose Input and Output Port 0.6 / External Interrupt 2/ SPI Chip Select 3 / UART RTS
47	RESET	Active Low. A low signal on this pin for 24 system clocks will cause the part to reset.
48	P4.0/PWM0	General Purpose Input and Output Port 4.0/PWM channel 0.
49	P4.1/PWM1	General Purpose Input and Output Port 4.1/PWM channel 1.
50	P4.2/PWM2	General Purpose Input and Output Port 4.2/PWM channel 2
51	P4.3/PWM3	General Purpose Input and Output Port 4.3/PWM channel 3.
52	P4.4/PWM4	General Purpose Input and Output Port 4.4/PWM channel 4.
53	P4.5/PWM5	General Purpose Input and Output Port 4.5/PWM channel 5.
54	LFXTAL2	32.768kHz watch crystal output for WU timers.
55	LFXTAL1	32.768kHz watch crystal input for WU timers.

**ADuCRF101 Pin Function Descriptions continued**

Pin No.	Pin Name	Description
56	VDD_DIG2	Voltage Regulator output for Digital. A 220 nF capacitor should be placed between this pin and ground for regulator stability and noise rejection.
57	VDDBAT	Battery Terminal, connected to the top die (UHF transceiver's LDOs) and to the LF receiver.
58	P4.6/PWM6	General Purpose Input and Output Port 4.6/PWM channel 6.
59	P4.7/PWM7	General Purpose Input and Output Port 4.7/PWM channel 7.
60	ADCVREF	Top die ADC Reference Output. A 220 nF capacitor should be placed between this pin and ground for adequate noise rejection.
61	P3.2/PWM_SYNC	General Purpose Input and Output Port 3.2/PWM synchronisation.
62	P3.3/PWM_TRIP	General Purpose Input and Output Port 3.3/PWM safety cut off.
63	P3.4	General Purpose Input and Output Port 3.4
64	P3.5	General Purpose Input and Output Port 3.5
65	PADDLE	Connected to the GROUND pin.

## Supplies and Grounds

See datasheet for pinout description.

The top die supply is VDDBAT1. The top die has 4 LDOs with a total of six outputs as follows: VDDRF1, VDDRF2, VDDVCO, VDDSYNTH, VDDDIG, VDDDIG. Ground is the paddle.

The bottom die supply is VBAT. The two IOVDD pins are used for the GPIOs only. VBAT is the supply for the regulators. The output of LDO is LVDD. The Cortex core, Flash etc are all powered by the regulated voltage from the internal LDO.

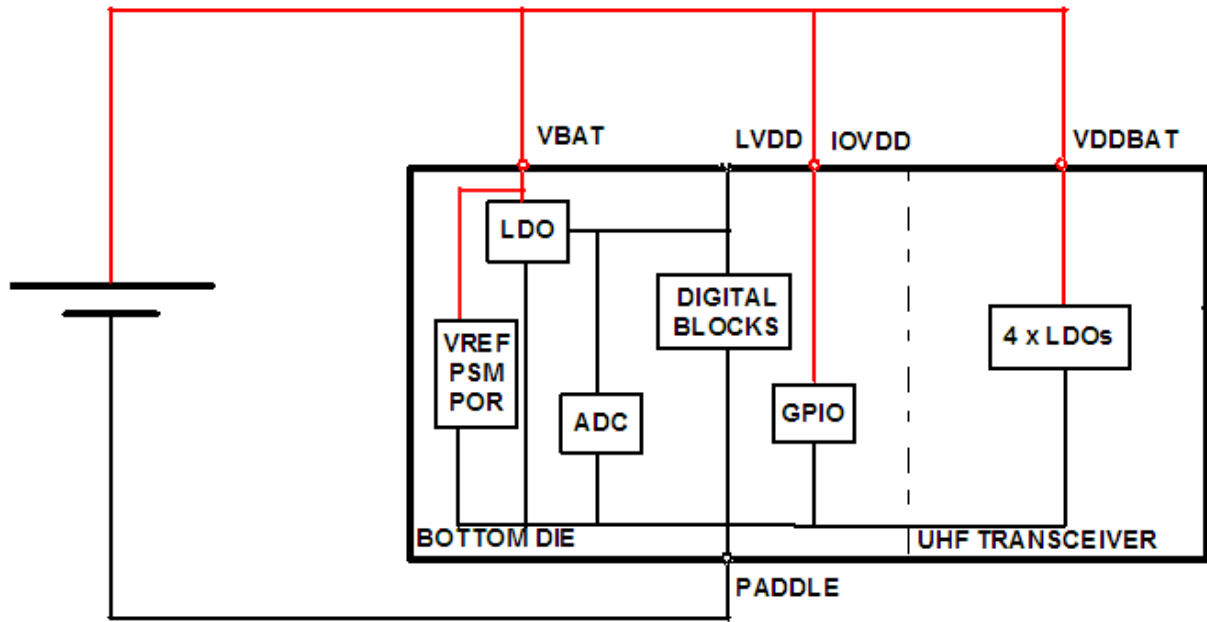


Figure 62: Simplified Supplies Connections

## Clocks

The ADuCRF101 has 4 clock sources, two internal and two external. Care must be taken when choosing the parallel resonant crystals.

### Low frequency crystal:

This crystal is mainly used to provide a high accuracy clock for the wakeup timer. The crystal oscillator is designed to support a 32.768kHz crystal or 65.536kHz crystal with maximum series resistance of 35kOhms and a load capacitance of 12.5pF, for example crystal from Citizen, series CFS or CFV.

The internal structure and connections are shown in Figure 63. Pin capacitance is approximately 2pF.

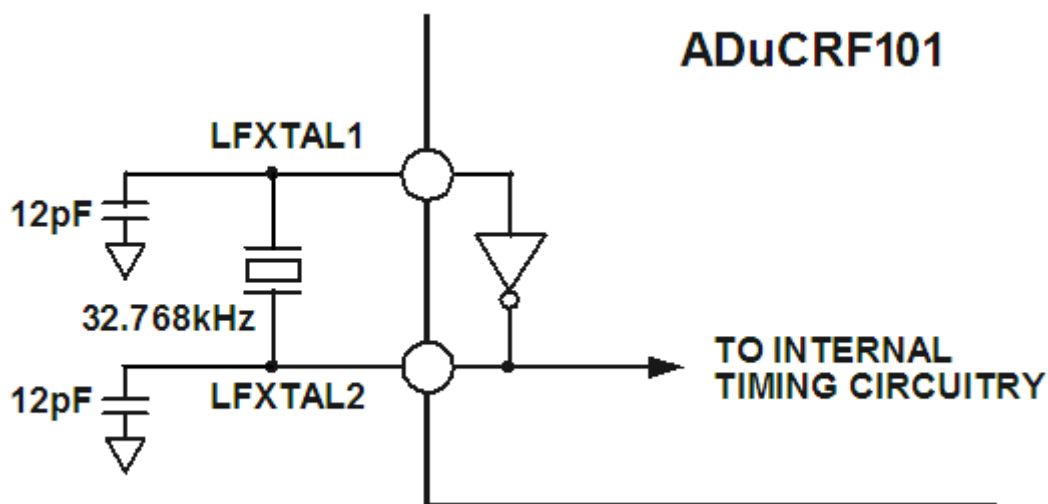






















Figure 63: External Parallel Resonant Crystal Connections

### High frequency crystal:

This crystal is used for the UHF transceiver.

## Serial Wire Debug Interface

Serial Wire Debug (SWD) provides a debug port for pin limited packages. SWD replaces the 5-pin JTAG port with a clock (SWDCLK) and a single bi-directional data pin (SWDIO), providing all the normal JTAG debug and test functionality. SWDIO and SWCLK are overlaid on the TMS and TCK pins on the ARM 20-PIN JTAG interface.

Vcc	1			2	Vcc(Optional)
NYU	3			4	GND
NYU	5			6	GND
SWDIO	7			8	GND
SWCLK	9			10	GND
NYU	11			12	GND
SWO	13			14	GND
RESET	15			16	GND
NYU	17			18	GND
NYU	19			20	GND

**Figure 64: SWD 20-PIN Connector Pinout**

**Table 320: SWD Connections**

Signal	Connect to...
SWDIO	Data I/O pin. Use 100K Ohm pull-up resistor to VCC.
SWO	No connect
SWCLK	Clock pin. Use 100K Ohm pull-up resistor to VCC
VCC	Positive Supply Voltage — Power supply for JTAG interface drivers.
GND	Digital ground.
RESET	No connect

## PA/LNA Matching

The UHF transceiver has a differential LNA and both a single-ended PA and differential PA. This flexibility allows numerous possibilities in interfacing the UHF transceiver to the antenna.

### Combined Single-Ended PA and LNA Match

This combined single-ended PA and LNA match allows the transmit and receive paths to be combined without the use of an external transmit/receive switch. The matching network design is shown in

Figure 65. The differential LNA match is a five element discrete balun giving a single-ended input. The single-ended PA output is a three element match consisting of the choke inductor to the CREGRF2 regulated supply and a series inductor and capacitor. The LNA and PA paths are combined and a T-stage harmonic filter provides attenuation of the transmit harmonics.

In a combined match, the off impedances of the PA and LNA need to be considered. This can lead to a small loss in transmit power and degradation in receiver sensitivity in comparison with a separate single-ended PA and LNA match. However, with optimum matching, the typical loss in transmit power is <0.5dB and the degradation in sensitivity is < 0.5dB when compared with a separate PA and LNA matching topology.

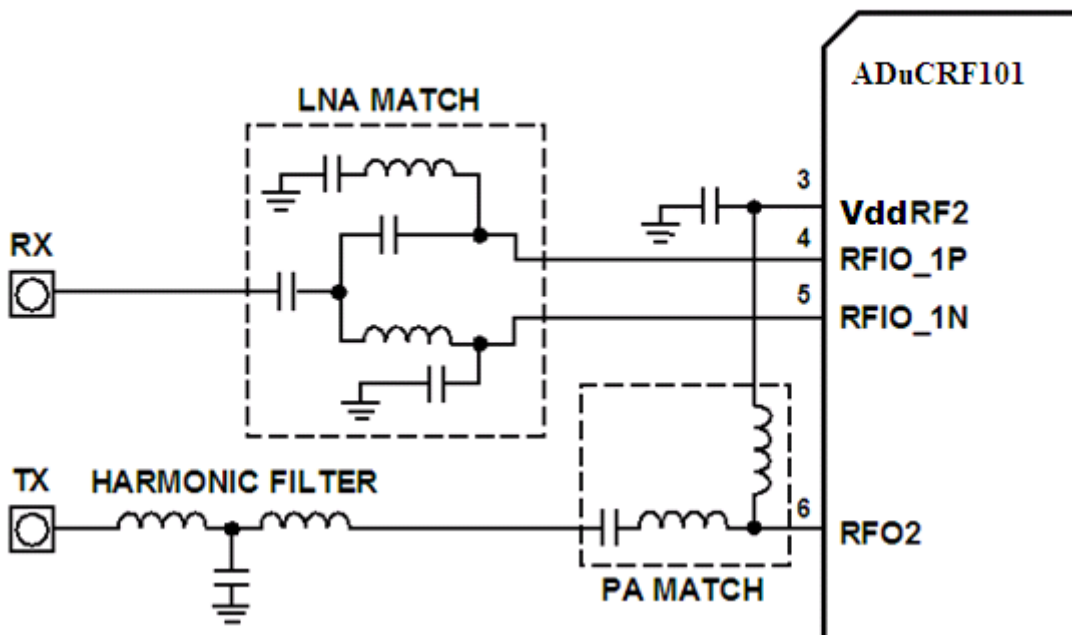


Figure 65: Combined single-ended PA and LNA match



## Separate Single-ended PA/LNA Match

The separate single-ended PA and LNA matching configuration is illustrated in Figure 66. The network is the same as the combined matching network shown in Figure 65 except the transmit and receive paths are separate. An external transmit/receive antenna switch can be used to combine the transmit and receive paths to allow connection to an antenna. In designing this matching network it is not necessary to consider the off impedances of the PA and LNA and thus achieving an optimum match is less complex than the combined single-ended PA and LNA match.

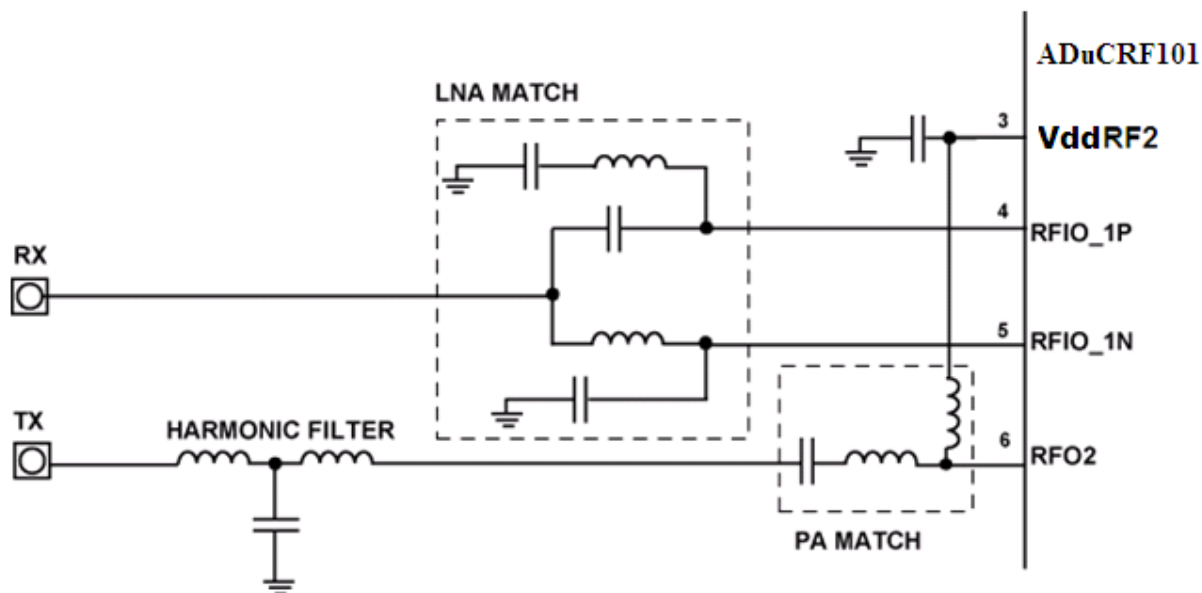


Figure 66: Separate Single-Ended PA and LNA Match

## Combined Differential PA/LNA Match

In this matching topology single ended PA is not used. The differential PA and LNA match comprises of a five element discrete balun giving a single-ended input/output as illustrated in Figure 67. The harmonic filter is used to minimize the RF harmonics from the differential PA.

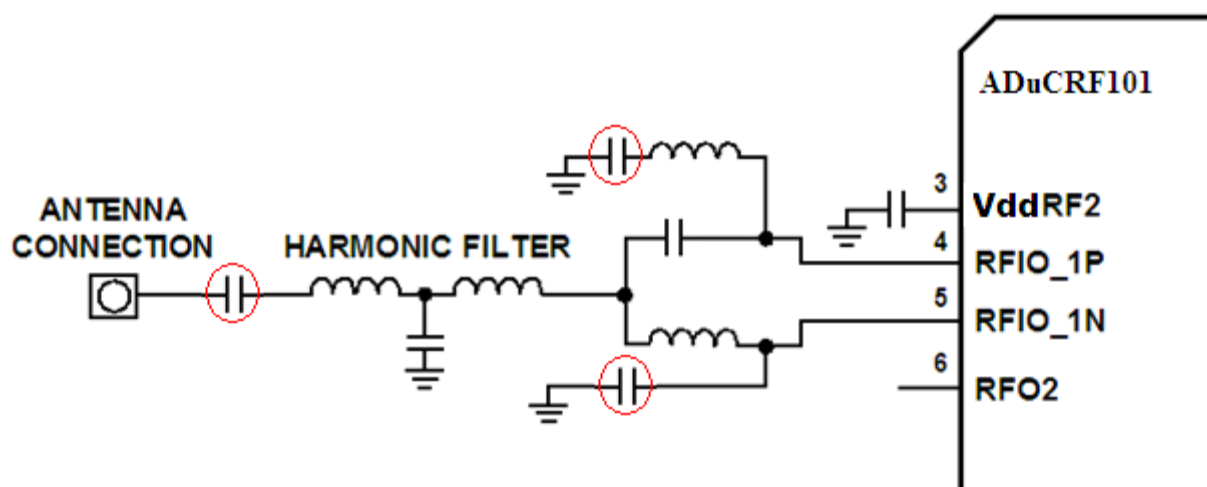


Figure 67: Combined Differential PA and LNA match

Note: The capacitor highlighted in the red circle should not exceed 100pF.

### Transmit Antenna Diversity

Transmit antenna diversity is possible using the differential PA and single ended PA. The required matching network is shown in Figure 68.

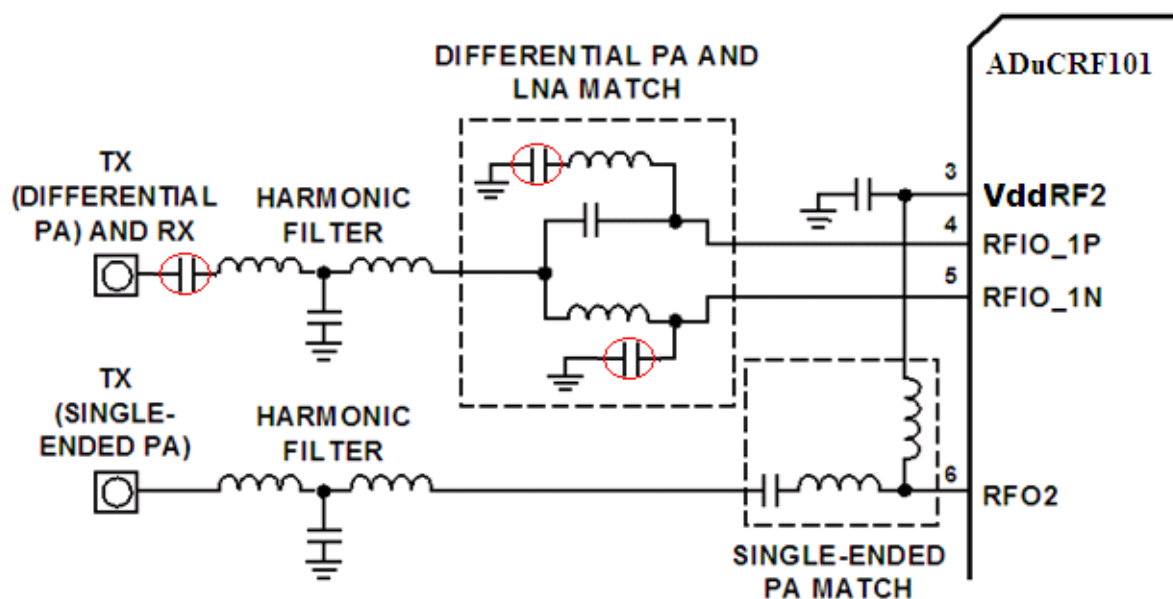


Figure 68: Matching Topology for Transmit Antenna Diversity

Note: The capacitor highlighted in the red circle should not exceed 100pF.