PRELIMINARY
Revision December 2005

# AR6001X ROCm™ Single-Chip MAC/BB/Radio for 2.4/5 GHz Embedded WLAN Applications

## General Description

The Atheros AR6001X is part of the AR6001 ROCm chipset family. It is a highly integrated, all-CMOS, single chip solution for combined cellular/handset applications and includes a 2.4/5 GHz radio, analog-to-digital and digital-to-analog converters, a baseband processor, multi-protocol media access control (MAC), and a MIPS CPU. It enables a high performance, cost effective, low power, compact solution in a dual-mode cellular/WLAN handset, PDA, VoIP handset, or MP3/4 player.

The AR6001X's transmitter combines baseband in-phase (I) and quadrature (Q) signals, converts them to the desired frequency, and drives the RF signal off-chip. The receiver uses an integrated dual-conversion architecture and requires no off-chip intermediate frequency (IF) filters. The frequency synthesizer supports one-MHz steps to match the frequencies defined by IEEE 802.11 specifications. All internal clocks are generated from a single external crystal.

The AR6001X implements half-duplex OFDM, CCK, and DSSS baseband processing supporting all IEEE 802.11a/g data rates. The MAC supports the IEEE 802.11 wireless MAC protocol as well as 802.11i security, receive and transmit filtering, error recovery, and quality of service (QoS).

The AR6001X provides multiple user interfaces including UART, SDIO or SPI, and $I^2C$. Other external interfaces include serial EEPROM, GPIOs, and LEDs.

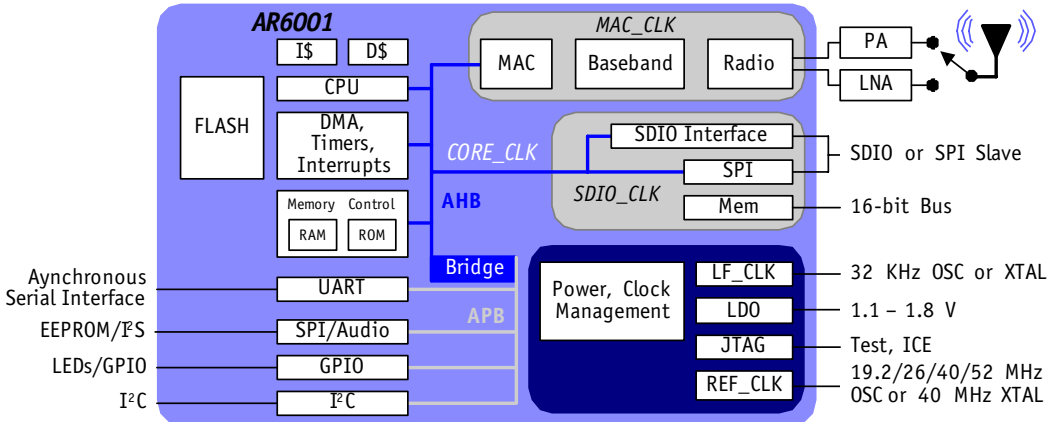
## AR6001X Features

- All-CMOS single chip for IEEE 802.11a/g compatible WLANs
- Operates in 2.4 and 5 GHz frequency bands.

| Freq | Bands | Frequency |
|---|---|---|
| 2.4 GHz | | 2.312–2.472 GHz, 2.484 GHz |
| 5 GHz | U-NII | 5.15–5.35 GHz, 5.725–5.825 GHz |
| | ISM | 5.725–5.850 GHz |
| | DSRC | 5.850–5.925 GHz |
| | Europe | 5.15–5.35 GHz, 5.47–5.725 GHz |
| | Japan | 4.90–5.00 GHz, 5.03–5.091 GHz, 5.15–5.25, 5.25–5.35 GHz |

- Data rates of 6–54 Mbps for 802.11a and 1–54 Mbps for 802.11g
- Integrated MIPS R4KEm CPU, clocked at up to 141 MHz
- 4 KB D-cache and 8 KB I-cache
- 80 KB on-chip SRAM, 256 KB on-chip ROM
- Stack-mounted 512 KB Flash
- UART and serial EEPROM
- Host interface support for SDIO/SPI, Local Bus, or 16-bit PC Card interface
- RTC support
- Sleep clock using 32 KHz clock
- Leading edge APSD support for energy efficient operation
- Bluetooth coexistence handshaking
- IEEE 1149.1 standard test access port and boundary scan architecture supported
- Advanced power management to minimize standby and active power
- Standard 0.18 µm CMOS technology
- 216-ball, 10 mm x 10 mm BGA package

## AR6001X ROCm Block Diagram



## Table of Contents


1 BGA Pin Descriptions 9
2 Functional Description 19
- 2.1 Overview 19
- 2.2 LEDs 19
- 2.3 Master SI/SPI Control 19
- 2.4 GPIO 19
  - 2.4.1 GPIO Pins 20
  - 2.4.2 Reading and Writing the GPIO pins 20
  - 2.4.3 Sigma Delta DACs for GPIOs 21
- 2.5 UART 21
- 2.6 Reset 21
  - 2.6.1 COLD_RESET 21
  - 2.6.2 WARM_RESET 21
  - 2.6.3 Reset Sequence 21
- 2.7 Power Management 22
  - 2.7.1 Hardware Power States 22
  - 2.7.2 Sleep State Management 23
- 2.8 System Clocking 23
  - 2.8.1 SLEEP_CLK 23
  - 2.8.2 REF_CLK 23
  - 2.8.3 Interface Clock 24
  - 2.8.4 Clock Generators 24
  - 2.8.5 Antenna Switching 24

3 Host Interfaces 27
- 3.1 SDIO/SPI Slave Interface 27
- 3.2 SDIO Address Map 27
- 3.3 SPI Interface 27
- 3.4 Local Bus Interface 27
- 3.5 Host Interface Address Map 27
- 3.6 Mailboxes 28
  - 3.6.6 Error Conditions 28
- 3.7 Interrupts 28
  - 3.7.1 AR6001X to Host 28
  - 3.7.2 Host to AR6001X 28
  - 3.7.3 SDIO Interface 28
  - 3.7.4 Local Bus Interface 29

4 Radio 31
- 4.1 Receiver (Rx) Block 31
- 4.2 Transmitter (TX) Block 32
  - 4.2.1 Synthesizer (SYNTH) Block 32
- 4.3 Bias/Control (BIAS) Block 32

5 Electrical Characteristics 33
- 5.1 Absolute Maximum Ratings 33
- 5.2 Recommended Operating Conditions 33
- 5.3 DC Electrical Characteristics 34
- 5.4 Radio Receiver Characteristics 37
- 5.5 Radio Transmitter Characteristics 39
- 5.6 AR6001X Synthesizer Characteristics 41
- 5.7 Power Consumption Parameters 42

6 AC Specifications 43
- 6.1 External 32 KHz Input Clock Timing 43
- 6.2 Local Bus Interface Timing 44
- 6.3 SD/SPI Interface Timing 45
- 6.4 IO Description 46
- 6.5 SPI Timing Flow 46
  - 6.5.1 PIO Writes 46
  - 6.5.2 PIO Reads 47
  - 6.5.3 DMA Writes 48
  - 6.5.4 DMA Reads 49
- 6.6 Error Recovery 50
- 6.7 Early Transaction Termination 50
- 6.8 Interrupts 50
- 6.9 32-Bit Operation 50
- 6.10 Clock Frequency Selection 51

7 Register Descriptions 53
- 7.1 RTC Block Registers 53
  - 7.1.1 Reset Control (RESET_CONTROL) 54
  - 7.1.2 Crystal Control (XTAL_CONTROL) 54
  - 7.1.3 TCXO Detection (TCXO_DETECT) 56
  - 7.1.4 PLL Control (PLL_CONTROL) 56
  - 7.1.5 PLL Settle Time (PLL_SETTLE) 56
  - 7.1.6 Crystal Settle Time (XTAL_SETTLE) 57
  - 7.1.7 Core Clock (CORE_CLOCK) 57
  - 7.1.8 CPU Clock (CPU_CLOCK) 58
  - 7.1.9 Clock Gating Control (CLOCK_CONTROL) 58
  - 7.1.10 Reference Voltage Trim Control (REF_VOLTAGE_TRIM) 58
  - 7.1.11 On-Chip LDO Control

(LDO_CONTROL) 59
7.1.12 Watchdog Timer (WDT_CONTROL) 59
7.1.13 Watchdog Timer Interrupt Status (WDT_STATUS) 59
7.1.14 Watchdog Timer Compare Target (WDT) 60
7.1.15 Watchdog Timer Current Count (WDT_COUNT) 60
7.1.16 Watchdog Timer Reset (WDT_RESET) 60
7.1.17 AR6001X CPU Interrupt Status (INT_STATUS) 61
7.1.18 LF Timer 0 Compare Target (LF_TIMER0) 61
7.1.19 LF Timer 0 Current Count (LF_TIMER_COUNT0) 62
7.1.20 LF Timer 0 Control Bits (LF_TIMER_CONTROL0) 62
7.1.21 LF Timer 0 Interrupt Status (LF_TIMER_STATUS0) 62
7.1.22 LF Timer 1 Compare Target (LF_TIMER1) 63
7.1.23 LF Timer 1 Current Count (LF_TIMER_COUNT1) 63
7.1.24 LF Timer 1 Control Bits (LF_TIMER_CONTROL1) 63
7.1.25 LF Timer 1 Interrupt Status (LF_TIMER_STATUS1) 64
7.1.26 LF Timer 2 Compare Target (LF_TIMER2) 64
7.1.27 LF Timer 2 Current Count (LF_TIMER_COUNT2) 64
7.1.28 LF Timer 2 Control Bits (LF_TIMER_CONTROL2) 65
7.1.29 LF Timer 2 Interrupt Status (LF_TIMER_STATUS2) 65
7.1.30 LF Timer 3 Compare Target (LF_TIMER3) 65
7.1.31 LF Timer 3 Current Count (LF_TIMER_COUNT3) 66
7.1.32 LF Timer 3 Control Bits (LF_TIMER_CONTROL3) 66
7.1.33 LF Timer 3 Interrupt Status (LF_TIMER_STATUS3) 66
7.1.34 HF Timer Compare Target (HF_TIMER) 66
7.1.35 HF Timer current count. (HF_TIMER_COUNT) 67
7.1.36 Captured LF Timer Value Relative to HF Timer Read (HF_LF_COUNT) 67
7.1.37 .HF Timer Control Bits (HF_TIMER_CONTROL) 67
7.1.38 HF Timer Interrupt Status (HF_TIMER_STATUS) 68
7.1.39 RTC Values Load into RTC Logic (RTC_CONTROL) 68
7.1.40 RTC Time of Day (RTC_TIME) 68
7.1.41 RTC Date and Year (RTC_DATE) 69
7.1.42 RTC Set Time of Day (RTC_SET_TIME) 69
7.1.43 RTC Set Date and Year (RTC_SET_DATE) 69
7.1.44 RTC Alarm Time of Day (RTC_SET_ALARM) 70
7.1.45 RTC Operation Configuration (RTC_CONFIG) 70
7.1.46 RTC Alarm Enable, Set and Clear (RTC_ALARM_STATUS) 71
7.1.47 UART Wakeup Events Enable (UART_WAKEUP) 71
7.1.48 Reset Cause (RESET_CAUSE) 71
7.1.49 System Sleep Status (SYSTEM_SLEEP) 73
7.1.50 LDO_D Voltage (LDO_VOLTAGE) 73
7.1.51 LDO_A Voltage (LDO_A_VOLTAGE) 74
7.1.52 SDIO_LDO voltage (SDIO_LDO_VOLTAGE) 74
7.1.53 Core Pad Enable (CORE_PAD_ENABLE) 75
7.1.54 SDIO Signal Wrapper (SDIO_WRAPPER) 75
7.1.55 MAC Sleep Options (MAC_SLEEP_CONTROL) 75
7.1.56 Keep Awake Timer (KEEP_AWAKE) 75
7.1.57 Chip Rev ID (CHIP_REV) 76
7.1.58 HF 32 KHz Clock Creation (DERIVED_RTC_CLK) 76
7.1.59 Automatic Clock Gating Control

(ACG_DISABLE) 76
7.2 Memory Block Registers 77
7.2.1 Bank 0 Address (BANK0_ADDR) 77
7.2.2 Bank 0 Configuration (BANK0_CONFIG) 78
7.2.3 Bank 0 Read Sequence (BANK0_READ) 79
7.2.4 Bank 0 Write Sequence (BANK0_WRITE) 80
7.2.5 Bank 1 Address (BANK1_ADDR) 81
7.2.6 Bank 1 Configuration (BANK1_CONFIG) 81
7.2.7 Bank 1 Read Sequence (BANK1_READ) 82
7.2.8 Bank 1 Write Sequence (BANK1_WRITE) 83
7.2.9 Bank 2 Address (BANK2_ADDR) 84
7.2.10 Bank 2 Configuration (BANK2_CONFIG) 84
7.2.11 Bank 2 Read Sequence (BANK2_READ) 85
7.2.12 Bank 2 Write Sequence (BANK2_WRITE) 86
7.2.13 Interrupt When Timing Margin Small (TIMING_INT_ENABLE) 87
7.2.14 MC Interrupt Bits Status (MC_ERROR_STATUS) 87
7.3 UART Registers 87
7.3.1 Receive Buffer (RBR) 88
7.3.2 Transmit Holding (THR) 88
7.3.3 Divisor Latch Low (DLL) 88
7.3.4 Divisor Latch High (DLH) 88
7.3.5 Interrupt Enable (IER) 89
7.3.6 Interrupt Identity (IIR) 89
7.3.7 FIFO Control (FCR) 89
7.3.8 Line Control (LCR) 90
7.3.9 Modem Control (MCR) 90
7.3.10 Line Status (LSR) 91
7.3.11 Modem Status (MSR) 92
7.4 Serial Interface Registers 92
7.4.1 SI Configuration (SI_CONFIG) 93
7.4.2 SI Control/Status (SI_CS) 94
7.4.3 First Four Bytes of Tx Data (SI_TXDATA0) 94
7.4.4 Second Four Bytes of Tx Data (SI_TXDATA1) 95
7.4.5 First Four Bytes of Rx Data (SI_RXDATA0) 95
7.4.6 Second Four Bytes of Rx Data (SI_RXDATA1) 95
7.5 GPIO Registers 96
7.5.1 Drive Data Out on GPIO Pins (GPIO_OUT) 96
7.5.2 Write 1 to Set GPIO_OUT Alias (GPIO_OUT_W1TS) 98
7.5.3 Write 1 to Clear GPIO_OUT Alias (GPIO_OUT_W1TC) 98
7.5.4 Enable Output Drivers for GPIO Pins (GPIO_ENABLE) 98
7.5.5 Write 1 to Set GPIO_ENABLE Alias (GPIO_ENABLE_W1TS) 98
7.5.6 Write 1 to Clear GPIO_ENABLE Alias (GPIO_ENABLE_W1TC) 99
7.5.7 Sample Data on GPIO Pins (GPIO_IN) 99
7.5.8 GPIO Pins Interrupt Status (GPIO_STATUS) 99
7.5.9 Write 1 to Set GPIO_STATUS Alias (GPIO_STATUS_W1TS) 99
7.5.10 Write 1 to Clear GPIO_STATUS Alias (GPIO_STATUS_W1TC) 100
7.5.11 GPIO 0 Configuration (GPIO_PIN0) 100
7.5.12 GPIO 1 Configuration (GPIO_PIN1) 101
7.5.13 GPIO 2 Configuration (GPIO_PIN2) 102
7.5.14 GPIO 3 Configuration (GPIO_PIN3) 103
7.5.15 GPIO 4 Configuration (GPIO_PIN4) 104
7.5.16 GPIO 5 Configuration (GPIO_PIN5) 105
7.5.17 GPIO 6 Configuration (GPIO_PIN6) 106
7.5.18 GPIO 7 Configuration

(GPIO_PIN7) 107
7.5.19 GPIO 8 Configuration (GPIO_PIN8) 108
7.5.20 GPIO 9 Configuration (GPIO_PIN9) 109
7.5.21 GPIO 10 Configuration (GPIO_PIN10) 110
7.5.22 GPIO 11 Configuration (GPIO_PIN11) 111
7.5.23 GPIO 12 Configuration (GPIO_PIN12) 112
7.5.24 GPIO 13 Configuration (GPIO_PIN13) 113
7.5.25 GPIO 14 Configuration (GPIO_PIN14) 114
7.5.26 GPIO 15 Configuration (GPIO_PIN15) 115
7.5.27 GPIO 16 Configuration (GPIO_PIN16) 116
7.5.28 GPIO 17 Configuration (GPIO_PIN17) 117
7.5.29 SDIO Pin Driver Configuration (SDIO_PIN) 118
7.5.30 CLK_REQ Pin Driver Configuration (CLK_REQ_PIN) 118
7.5.31 Sigma Delta PWM Configuration (SIGMA_DELTA) 119
7.6 AR6001X Side MBOX and Host IF Registers 119
7.6.1 MBOX PIO Access (MBOX_FIFO) 120
7.6.2 Non-Destructive FIFO Status Query (MBOX_FIFO_STATUS) 121
7.6.3 MBOX DMA Engine Policy Control (MBOX_DMA_POLICY) 121
7.6.4 MBOX 0 Rx DMA Descriptor Base Address (MBOX0_DMA_RX_DESCRIPTOR_BASE) 122
7.6.5 MBOX 0 Rx DMA Control (MBOX0_DMA_RX_CONTROL) 122
7.6.6 MBOX 0 Tx DMA Descriptor Base Address (MBOX0_DMA_TX_DESCRIPTOR_BASE) 123
7.6.7 MBOX 0 Tx DMA Control (MBOX0_DMA_TX_CONTROL) 123
7.6.8 MBOX 1 Rx DMA Descriptor Base Address (MBOX1_DMA_RX_DESCRIPTOR_BASE) 123
7.6.9 MBOX 1 Rx DMA Control (MBOX1_DMA_RX_CONTROL) 123
7.6.10 MBOX 1 Tx DMA Descriptor Base Address (MBOX1_DMA_TX_DESCRIPTOR_BASE) 124
7.6.11 MBOX 1 Tx DMA Control (MBOX1_DMA_TX_CONTROL) 124
7.6.12 MBOX 2 Rx DMA Descriptor Base Address (MBOX2_DMA_RX_DESCRIPTOR_BASE) 124
7.6.13 MBOX 2 Rx DMA Control (MBOX2_DMA_RX_CONTROL) 124
7.6.14 MBOX 2 Tx DMA Descriptor Base Address (MBOX2_DMA_TX_DESCRIPTOR_BASE) 124
7.6.15 MBOX 2 Tx DMA Control (MBOX2_DMA_TX_CONTROL) 125
7.6.16 MBOX 3 Rx DMA Descriptor Base Address (MBOX3_DMA_RX_DESCRIPTOR_BASE) 125
7.6.17 MBOX 3 Rx DMA Control (MBOX3_DMA_RX_CONTROL) 125
7.6.18 MBOX 3 Tx DMA Descriptor Base Address (MBOX3_DMA_TX_DESCRIPTOR_BASE) 126
7.6.19 MBOX 3 Tx DMA Control (MBOX3_DMA_TX_CONTROL) 126
7.6.20 MBOX-Related Interrupt Status (MBOX_INT_STATUS) 126
7.6.21 MBOX-Related Interrupt Enables (MBOX_INT_ENABLE) 127

7.6.22 Host CPU Interrupt (INT_HOST) 128
7.6.23 Credit Counters Direct Access (LOCAL_COUNT) 128
7.6.24 Credit Counter Atomic Increment (COUNT_INC) 128
7.6.25 Interface Scratch (LOCAL_SCRATCH) 128
7.6.26 LB Configuration (USE_LOCAL_BUS) 129
7.6.27 SDIO Configuration (SDIO_CONFIG) 129
7.6.28 Stereo Block Configuration (STEREO_CONFIG) 129
7.6.29 Set Stereo Volume (STEREO_VOLUME) 130
7.6.30 Host Interface Access (HOST_IF_WINDOW) 131
7.7 Host Interface Registers 131
7.7.1 Pending Interrupt Status (HOST_INT_STATUS) 133
7.7.2 CPU-Sourced Interrupt Status (CPU_INT_STATUS) 133
7.7.3 Error or Wakeup Interrupt Status (ERROR_INT_STATUS) 133
7.7.4 Host IF Credit Counter Interrupt (COUNTER_INT_STATUS) 134
7.7.5 Mailbox FIFO Status (MBOX_FRAME) 134
7.7.6 Valid Bits for Lookahead (RX_LOOKAHEAD_VALID) 134
7.7.7 Lookahead to Next 4 MBOX Rx0 FIFO Bytes (RX_LOOKAHEAD0) 134
7.7.8 Lookahead to Next 4 MBOX Rx1 FIFO Bytes (RX_LOOKAHEAD1) 136
7.7.9 Lookahead to Next 4 MBOX Rx2 FIFO Bytes (RX_LOOKAHEAD2) 136
7.7.10 Lookahead to Next 4 MBOX Rx3 FIFO Bytes (RX_LOOKAHEAD3) 136
7.7.11 Credit Counters Direct Access (COUNT) 136
7.7.12 Credit Counter Atomic Decrement (COUNT_DEC) 136
7.7.13 Interface Scratch (SCRATCH) 137
7.7.14 HOST_INT_STATUS Enable Bits (INT_STATUS_ENABLE) 137
7.7.15 CPU Sourced Interrupt Status (CPU_INT_STATUS_ENABLE) 137
7.7.16 Error Interrupt Status (ERROR_STATUS_ENABLE) 137
7.7.17 Credit Counter Interrupt Status (COUNTER_INT_STATUS_ENABLE) 138
7.7.18 FIFO Timeout Period (FIFO_TIMEOUT) 138
7.7.19 FIFO Timeout Enable. (FIFO_TIMEOUT_ENABLE) 138
7.7.20 Disable Sleep Mode (DISABLE_SLEEP) 138
7.7.21 LB Endianness (LOCAL_BUS_ENDIAN) 139
7.7.22 LB and SPI Host Interface State (LOCAL_BUS) 139
7.7.23 AR6001X CPU Interrupt (INT_WLAN) 139
7.7.24 SPI Slave Interface Configuration (SPI_CONFIG) 141
7.7.25 SPI Status (SPI_STATUS) 142
7.7.26 SDIO CIS Tuples Copy (CIS_WINDOW) 143
8 Package Dimensions 145
9 Ordering Information 147

# 1. BGA Pin Descriptions

This section contains a listing of the signal descriptions (see Table 1-1 for BGA package pins).

The following nomenclature is used for signal names:

- NC indicates no connection should be made to this pin.
- _L at the end of the signal name indicates active low signals.
- P at the end of the signal name indicates the positive side of a differential signal.
- N at the end of the signal name indicates the negative side of a differential signal.

The following nomenclature is used for signal types described in Table 1-4:

- IA indicates an analog input signal.
- I indicates a digital input signal.
- IH indicates input signals with weak internal pull-up, to prevent signals from floating when left open.
- IL indicates input signals with weak internal pull-down, to prevent signals from floating when left open.
- I/O indicates a digital bidirectional signal.
- OA indicates an analog output signal.
- O indicates a digital output signal.
- P indicates a power or ground signal.

*Table 1-1.* **BGA Package Pinout**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | AGND | AVDD33 | AVDD33 | AVDD33 | Table 1-3 | RF2 OUTP | RF2 OUTN | RF5 OUTP | RF5 OUTN | AGND | XTALI | AVDD33 | ANTD | ANTC | ANTB | ANTE | LDO_ BYPASS | GND |
| B | AVDD18 | AGND | AVDD33 | AVDD33 | XPABIAS 5 | PA2 BIASP | PA2 BIASN | PA5 BIASP | PA5 BIASN | AGND | XTAL0 | AVDD33 | ANTA | DVDD3_ ANT | DVDD3_ ANT | DVDD3_ ANT | GND | Table 1-3 |
| C | AVDD18 | AVDD18 | AGND | AVDD33 | AVDD33 | PDET | AVDD18 | AVDD18 | AGND | AGND | AVDD18 | AVDD18 | AVDD33 | DVDD 18 | DVDD 18 | GND | DVDD3 | Table 1-3 |
| D | RF5INN | RF5INP | AVDD33 | | | | | | | | | | | | | DVDD3 | DVDD3 | Table 1-3 |
| E | RF2INN | RF2INP | AVDD33 | | | | | | | | | | | | | DVDD3 | DVDD3 | Table 1-3 |
| F | NC | NC | AVDD18 | | | | | | | | | | | | | Table 1-3 | Table 1-3 | Table 1-3 |
| G | NC | NC | AVDD18 | | | | AGND | AGND | AGND | AGND | GND | GND | | | | DVDD 18 | Table 1-3 | Table 1-3 |
| H | BIAS REF | AVDD18 | AVDD18 | | | | AGND | AGND | AGND | AGND | GND | GND | | | | DVDD 18 | Table 1-3 | Table 1-3 |
| J | AVDD33 | AGND | GND | | | | AGND | AGND | AGND | AGND | GND | GND | | | | GND | DVDD3_ MEM | LB_ DATA_9 |
| K | LF XTALI | DVDD 18 | GND | | | | AGND | AGND | AGND | AGND | GND | GND | | | | GND | GND | LB_DAT A_10 |
| L | LF XTAL0 | WE_L | DVDD 18 | | | | GND | GND | GND | GND | GND | GND | | | | DVDD 18 | LB_ DATA_ 11 | MEM_ OE_L |
| M | RY/BY_L | RESET_L | DVDD 18 | | | | GND | GND | GND | GND | GND | GND | | | | DVDD 18 | OE_L | LB_ DATA_ 12 |
| N | MEM_ WE_L | RST_ OUT_L | DVDD 18 | | | | | | | | | | | | | Table 1-3 | Table 1-3 | Table 1-3 |
| P | DVDD 18_SDIO | DVDD 18_SDIO | DVDD3 | | | | | | | | | | | | | DVDD3 | DVDD3 | LB_ DATA_4 |
| R | Table 1-3 | Table 1-3 | DVDD3 | | | | | | | | | | | | | DVDD3 | DVDD3 | Table 1-3 |
| T | Table 1-3 | Table 1-3 | GND | DVDD3 | DVDD3 | CHIP_ PWD | DVDD 18 | DVDD 18 | GND | GND | DVDD 18 | DVDD 18 | LB_ BEO_L | DVDD3 | DVDD3 | GND | LB_ DATA_2 | CE_L |
| U | Table 1-3 | GND | SYS_ RST_L | Table 1-3 | DVDD3_ SDIO | TDO | TRST_L | TMS | GND | LB_ BE1_L | Table 1-3 | LB_ OE_L | LB_ ADDR_0 | LB_ ADDR_1 | LB_ ADDR_4 | LB_ ADDR_3 | GND | LB_ DATA_3 |
| V | GND | Table 1-3 | CLK_ REQ | SDIO_ LDO_BY PASS | DVDD3_ SDIO | NC | EJTAG_ SEL | TCK | TDI | LB_ WE_L | Table 1-3 | LB_ REG_L | LB_ CS_L | LB_ ADDR_2 | LB_ DATA_0 | MEM_ CS_0_L | LB_ DATA_1 | GND |

The AR6001's interface bus can be configured to be in SPI, SDIO, or Local Bus mode. Table 1-2 shows pin settings for mode configuration using the GPIO9 and TDO pins sampled during reset.

After reset, the particular mode set by these pins is not changeable by software.

When the AR6001 is configured to SDIO or SPI mode, the interface pins assume functionality as listed under the Default column of Table 1-3.

Refer to "GPIO Registers" on page 96 for configuration details of the GPIO pins.

*Table 1-2.* **Pin Settings for Mode Configuration**

| GPIO9 | TDO | Configuration |
|---|---|---|
| 0 | 0 | Generic SPI Mode |
| 0 | 1 | SDIO Mode (Default, GPIO9 pin has weak internal pull down, but TDO pin must be pulled high on the board) |
| 1 | 0 | Local Bus Mode |
| 1 | 1 | Reserved |

Table 1-3 shows the BGA multiplex pins (pins that share functions).

*Table 1-3.* **BGA Package Multiplexed Pins**

| Pin | Local Bus Mode | SDIO/SPI Mode | | |
|---|---|---|---|---|
| | | Default | GPIO | SPI Master |
| B18 | LB_DATA_14 | I2S_WS | GPIO10 | |
| C18 | LB_DATA_13 | I2S_SD | GPIO11 | |
| D18 | uses I2C_SCL0 | I2C_SCL0 | GPIO0 | SPI_CK |
| E18 | uses I2C_SDA0 | I2C_SDA0 | GPIO1 | SPI_MISO |
| F16 | uses BT_ACTIVE | BT_ACTIVE | GPIO4 | |
| F17 | uses RXD0 | RXD0 | GPIO3 | SPI_MOSI |
| F18 | uses TXD0 | TXD0 | GPIO2 | SPI_CS0_L |
| G17 | uses RX_CLEAR | RX_CLEAR | GPIO6 | |
| G18 | uses BT_PRIORITY | BT_PRIORITY | GPIO5 | |
| H17 | LB_DATA_15 | I2S_MCK | GPIO8 | |
| H18 | uses BT_FREQ | BT_FREQ | GPIO7 | |
| N16 | LB_DATA_7 | LB_DATA_7 | GPIO17 | Reserved |
| N17 | LB_DATA_6 | LB_DATA_6 | GPIO16 | Reserved |
| N18 | LB_DATA_8 | LB_DATA_8 | GPIO12/UART_CTS_L[1] | SPI_CS0_L[2] |
| R1 | LB_ADDR_5 | SDIO_CMD | | G_SPI_MOSI |
| R2 | LB_ADDR_7 | SDIO_DATA_2 | | |
| R18 | LB_DATA_5 | LB_DATA_5 | GPIO15 | Reserved |
| T1 | LB_ADDR_6 | SDIO_DATA_3 | | G_SPI_CS |
| T2 | LB_ADDR_8 | SDIO_DATA_1 | | G_SPI_INT |
| U1 | LB_ADDR_9 | SDIO_DATA_0 | | G_SPI_MISO |
| U4 | uses I2S_CK | I2S_CK | GPIO9 | |
| U11 | LB_INT_L | LB_INT_L | GPIO14/UART_CLK[3] | |
| V2 | LB_ADDR_10 | SDIO_CLK | | G_SPI_CLK |
| V11 | LB_WAIT_L | LB_WAIT_L | GPIO13/UART_RTS_L[4] | SPI_MOSI[2] |

[1]See the register "GPIO 12 Configuration (GPIO_PIN12)" on page 112 for more information.
[2]Duplicate for flexibility.
[3]See the register "GPIO 14 Configuration (GPIO_PIN14)" on page 114 for more information.
[4]See the register "GPIO 13 Configuration (GPIO_PIN13)" on page 113 for more information.

*Table 1-4.* **BGA Package Signal to Pin Relationships and Descriptions**

| Symbol | Pin | Type | Source or Destination | External PAD Power | Description |
|---|---|---|---|---|---|
| **Radio** | | | | | |
| ANTA | B13 | O | Antenna | DVDD3_ANT | LNA switch biasing. Output to control antenna switching. |
| ANTB | A15 | O | Antenna | DVDD3_ANT | |
| ANTC | A14 | O | Antenna | DVDD3_ANT | RF switch/ antenna biasing. Output to control antenna switching. |
| ANTD | A13 | O | Antenna | DVDD3_ANT | |
| ANTE | A16 | O | Antenna | DVDD3_ANT | |
| BIASREF | H1 | IA | — | — | Connects a 6.19KΩ ± 1% resistor to ground |
| PDET | C6 | IA | Power detector | — | Power detector signal. Voltage range and input impedence is TBD. |
| RF2INN | E1 | IA | RF input | — | Differential RF inputs at 2.4/5 GHz. Use one side for single-ended input. |
| RF2INP | E2 | IA | RF input | — | |
| RF5INN | D1 | IA | RF input | — | |
| RF5INP | D2 | IA | RF input | — | |
| RF2OUTP | A6 | OA | RF output | — | Differential RF power amplifier output |
| RF2OUTN | A7 | OA | RF output | — | |
| RF5OUTN | A9 | OA | RF output | — | |
| RF5OUTP | A8 | OA | RF output | — | |
| PA2BIASN | B7 | P | DC input | — | Biasing for differential radio output power. Connect through an external 2 nH for 2.4 GHz and 1 nH for 5 GHz, to AVDD18. Can use printed traces instead of the discrete inductors. |
| PA2BIASP | B6 | P | DC input | — | |
| PA5BIASN | B9 | P | DC input | — | |
| PA5BIASP | B8 | P | DC input | — | |
| XPABIAS2 | A5 | OA | XPA | — | Biasing external 2.4 GHz PA (analog). Programmable regulated voltage in 0.1 V steps.<br>This table depicts the XPABIAS2 voltage level over the range of the AVDD3_LDO power supply (2.9V to 3.6V) for the various programmed values. |
| XPABIAS5 | B5 | OA | XPA | — | Biasing external 5 GHz PA (analog). Programmable regulated voltage in 0.1 V steps.<br>This table depicts the XPABIAS5 voltage level over the range of the AVDD3_LDO power supply (2.9V to 3.6V) for the various programmed values. |

| AVDD3_LDO | Programmed Values | | | |
|---|---|---|---|---|
| | 2.7 V | 2.8 V | 2.9 V | 3.0 V |
| >3.1 V | 2.7 V | 2.8 V | 2.9 V | 3.0 V |
| 3.0 V | 2.7 V | 2.8 V | 2.9 V | >2.9 V |
| 2.9 V | 2.7 V | 2.8 V | >2.8 V | >2.8 V |

| AVDD3_LDO | Programmed Values | | | |
|---|---|---|---|---|
| | 2.7 V | 2.8 V | 2.9 V | 3.0 V |
| >3.1 V | 2.7 V | 2.8 V | 2.9 V | 3.0 V |
| 3.0 V | 2.7 V | 2.8 V | 2.9 V | >2.9 V |
| 2.9 V | 2.7 V | 2.8 V | >2.8 V | >2.8 V |

*Table 1-4.* **BGA Package Signal to Pin Relationships and Descriptions (continued)**

| Symbol | Pin | Type | Source or Destination | External PAD Power | Description |
|---|---|---|---|---|---|
| **Clock** | | | | | |
| LFXTALI | K1 | Crystal input | 32 KHz crystal | — | Low frequency clock (32 KHz crystal or oscillator required) |
| LFXTALO | L1 | Crystal output | 32 KHz crystal | — | |
| XTALO | B11 | Crystal output | 40 MHz crystal or external oscillator | — | High frequency clock |
| XTALI | A11 | Crystal input | 40 MHz crystal | — | |
| CLK_REQ | V3 | I/O | — | DVDD3_SDIO | When no crystal is connected to the AR6001X, this pin asserts when the AR6001X requires the 40 MHz clock. The host chip must provide a clock to the AR6001X within 2 ms after asserting CLK_REQ. CLK_REQ is a boot configuration pin on reset and is latched when SYS_RST_L is deasserted.<br>■ 0 = The AR6001X enters WLAN_OFF state on SYS_RST_L de-assertion<br>■ 1 = The AR6001X enters SDIO_ON state on SYS_RST_L de-assertion (for debug) |
| **Digital Control** | | | | | |
| SYS_RST_L | U3 | IH | — | DVDD3_SDIO | AR6001X Reset; must be asserted when power first applied to the chip, then released before any transactions can start |
| CHIP_PWD (PAD_DISABLE) | T6 | I | — | DVDD3_SDIO | ■ When LDO_BYPASS is disabled: Asserting this bump powers down the AR6001X to minimal power. No LDO is enabled, no state retained, and no transactions performed while CHIP_PWD is asserted. When de-asserting CHIP_PWD, SYS_RST_L must be asserted to restart the AR6001X.<br>■ When LDO_BYPASS is enabled: Asserting this bump only disables the digital PADs on the chip. |
| LDO_BYPASS | A17 | IL | — | DVDD3_ANT | Assert to bypass on-chip LDO for the digital core (except the SDIO block). If bypassed, the board must supply 1.8 V to the digital core via the DVDD18 pins and AVDD18. |
| BT_ACTIVE | F16[1] | I | — | DVDD3 | Indicates medium busy from an external source; can be asserted (e.g. by a Bluetooth device) to prevent the AR6001X from transmitting a new frame. Tie to ground when not in use. |
| BT_FREQ | H18[1] | I | — | DVDD3 | (optional) Indicates external source is transmitting on a restricted frequency band. Tie to ground when not in use. |
| BT_PRIORITY | G18[1] | I | — | DVDD3 | (optional) When BT_ACTIVE is asserted, indicates the external device transmits or receives at high priority. Tie to ground when not in use. |
| RX_CLEAR | G17[1] | O | — | DVDD3 | Indicates medium clear to an external device (e.g., Bluetooth), which should transmit only when RXCLEAR is asserted |

*Table 1-4.* **BGA Package Signal to Pin Relationships and Descriptions (continued)**

| Symbol | Pin | Type | Source or Destination | External PAD Power | Description |
|---|---|---|---|---|---|
| **I²C** | | | | | |
| I2C_SCL0 | D18[1] | I | — | DVDD3 | I²C Bus |
| I2C_SDA0 | E18[1] | O | — | DVDD3 | |
| **I²S** | | | | | |
| I2S_CK | U4[1] | I/O | — | DVDD3_SDIO | I²S Bus. Digital audio interface. |
| I2S_MCK | H17[1] | I/O | — | DVDD3 | |
| I2S_WS | B18[1] | I/O | — | DVDD3 | |
| I2S_SD | C18[1] | I/O | — | DVDD3 | |
| **UART** | | | | | |
| RXD0 | F17[1] | I | — | DVDD3 | UART receive and transmit data, compatible to 16550 |
| TXD0 | F18[1] | O | — | DVDD3 | |
| UART_CLK | U11[1] | I | — | | (optional) UART clock |
| UART_CTS_L | N18[1] | I | — | | (optional) UART Clear to Send |
| UART_RTS_L | V11[1] | O | — | | (optional) UART Request to Send |
| **SPI Master** | | | | | |
| SPI_CK | D18[1] | I/O | — | DVDD3 | SPI master clock |
| SPI_CS0_L | F18 or N18[1] | I/O | — | DVDD3 | SPI master chip select |
| SPI_MISO | E18[1] | I/O | — | DVDD3 | SPI master in/slave out |
| SPI_MOSI | F17 or V11[1] | I/O | — | DVDD3 | SPI master out/slave in |
| **Local Bus** | | | | | |
| LB_ADDR_0 | U13 | I | Local Bus | DVDD3 | Address [0] |
| LB_ADDR_1 | U14 | I | Local Bus | DVDD3 | Address [1] |
| LB_ADDR_2 | V14 | I | Local Bus | DVDD3 | Address [2] |
| LB_ADDR_3 | U16 | I | Local Bus | DVDD3 | Address [3] |
| LB_ADDR_4 | U15 | I | Local Bus | DVDD3 | Address [4] |
| LB_ADDR_5 | R1[1] | I | Local Bus | DVDD3_SDIO | Address [5] |
| LB_ADDR_6 | T1[1] | I | Local Bus | DVDD3_SDIO | Address [6] |
| LB_ADDR_7 | R2[1] | I | Local Bus | DVDD3_SDIO | Address [7] |
| LB_ADDR_8 | T2[1] | I | Local Bus | DVDD3_SDIO | Address [8] |
| LB_ADDR_9 | U1[1] | I | Local Bus | DVDD3_SDIO | Address [9] |
| LB_ADDR_10 | V2[1] | I | Local Bus | DVDD3_SDIO | Address [10] |
| LB_BE0_L | T13 | I | Local Bus | DVDD3 | In Local Bus mode, these pins are used as active-low byte enables. In PC Card mode, these pins are used as active-low byte chip selects. In both cases, LB_BE_L[0] refers to bits [7:0] of the data and LB_BE_L[1] refers to data bits [15:8]. |
| LB_BE1_L | U10 | I | Local Bus | DVDD3 | |
| LB_CS_L | V13 | I | Local Bus | DVDD3 | Local bus chip select |

*Table 1-4.* **BGA Package Signal to Pin Relationships and Descriptions (continued)**

| Symbol | Pin | Type | Source or Destination | External PAD Power | Description |
|---|---|---|---|---|---|
| LB_DATA_0 | V15 | I/O | Local Bus | DVDD3 | Data Bus. For eight-bit reads, unselected parts of the bus are not driven. The bus starts up as little Endian, but can be configured using a register write for big Endian operation. |
| LB_DATA_1 | V17 | I/O | Local Bus | DVDD3 | |
| LB_DATA_2 | T17 | I/O | Local Bus | DVDD3 | |
| LB_DATA_3 | U18 | I/O | Local Bus | DVDD3 | |
| LB_DATA_4 | P18 | I/O | Local Bus | DVDD3 | |
| LB_DATA_5 | R18[1] | I/O | Local Bus | DVDD3 | |
| LB_DATA_6 | N17[1] | I/O | Local Bus | DVDD3 | |
| LB_DATA_7 | N16[1] | I/O | Local Bus | DVDD3 | |
| LB_DATA_8 | N18[1] | I/O | Local Bus | DVDD3 | |
| LB_DATA_9 | J18[1] | I/O | Local Bus | DVDD3 | |
| LB_DATA_10 | K18[1] | I/O | Local Bus | DVDD3 | |
| LB_DATA_11 | L17[1] | I/O | Local Bus | DVDD3 | |
| LB_DATA_12 | M18[1] | I/O | Local Bus | DVDD3 | |
| LB_DATA_13 | C18[1] | I/O | Local Bus | DVDD3 | |
| LB_DATA_14 | B18[1] | I/O | Local Bus | DVDD3 | |
| LB_DATA_15 | H17[1] | I/O | Local Bus | DVDD3 | |
| LB_INT_L | U11[1] | I/O | Local Bus | DVDD3 | On bootup, the AR6001X is in memory mode and uses this pin to indicate whether or not it is ready (high equals ready). If the host configures the part for IO Mode by writing to the CCR register (in the config address space) or the standard CCR register (in the standard address space), then this pin is converted to an active-low level-triggered interrupt. |
| LB_OE_L | U12 | I/O | Local Bus | DVDD3 | Output enable |
| LB_REG_L | V12 | I/O | Local Bus | DVDD3 | ■ If set, reads and writes access configuration memory space instead of the standard memory space. In the configuration memory space, the CIS Tuples (located at address 0x600 of the standard space) are located at address 0x0. The CCR register is also accessible at address 0x200. In this mode, only byte reads to even addresses are supported.<br>■ If clear, all addresses are accessed as described in the standard address space. |
| LB_WAIT_L | V11[1] | I/O | Local Bus | DVDD3 | Asserted to keep the host from deasserting LB_OE_L or LB_WE_L. On reads, used to wait for the read data to be valid. On writes, used to postpone new requests until the previous write finishes. |
| LB_WE_L | V10 | I/O | Local Bus | DVDD3 | Write enable |
| **SDIO** | | | | | |
| SDIO_CLK | V2[1] | I | — | DVDD3_SDIO | SDI input clock from host (up to 25 MHz) |
| SDIO_CMD | R1[1] | I | — | DVDD3_SDIO | SDIO command line |
| SDIO_DATA_0 | U1[1] | I/O | — | DVDD3_SDIO | SDIO data lines |
| SDIO_DATA_1 | T2[1] | I/O | — | DVDD3_SDIO | |
| SDIO_DATA_2 | R2[1] | I/O | — | DVDD3_SDIO | |
| SDIO_DATA_3 | T1[1] | I/O | — | DVDD3_SDIO | |

*Table 1-4.* **BGA Package Signal to Pin Relationships and Descriptions (continued)**

| Symbol | Pin | Type | Source or Destination | External PAD Power | Description |
|---|---|---|---|---|---|
| SDIO_LDO_BYPASS | V4 | IL | — | DVDD3_SDIO | Assert to bypass on-chip LDO for the SDIO block. If bypassed, the board must supply 1.8 V to the SDIO block via the DVDD18_SDIO pins. |
| **SPI Slave** | | | | | |
| G_SPI_CS | T1[1] | I/O | — | DVDD3_SDIO | SPI slave chip select |
| G_SPI_MOSI | R1[1] | I | — | DVDD3_SDIO | SPI slave input master output |
| G_SPI_MISO | U1[1] | I/O | — | DVDD3_SDIO | SPI slave output master input |
| G_SPI_CLK | V2[1] | I | — | DVDD3_SDIO | SPI slave clock |
| G_SPI_INT | T2[1] | I/O | — | DVDD3_SDIO | SPI interrupt from AR6001X to host |
| **GPIO** | | | | | |
| GPIO0 | D18[1] | I/OL | — | DVDD3 | General purpose I/O [3-0]. Default to inputs, control using the GPIOCR register.<br>Input from the GPIOs can be read using the GPIODI register. Output to the GPIOs is provided by the GPIODO register. |
| GPIO1 | E18[1] | I/OL | — | DVDD3 | |
| GPIO2 | F18[1] | I/OL | — | DVDD3 | |
| GPIO3 | F17[1] | I/OL | — | DVDD3 | |
| GPIO4 | F16[1] | I/OL | — | DVDD3 | |
| GPIO5 | G18[1] | I/OL | — | DVDD3 | |
| GPIO6 | G17[1] | I/OL | — | DVDD3 | |
| GPIO7 | H18[1] | I/OL | — | DVDD3 | |
| GPIO8 | H17[1] | I/OL | — | DVDD3 | |
| GPIO9 | U4[1] | I/OL | — | DVDD3_SDIO | |
| GPIO10 | B18[1] | I/OL | — | DVDD3 | |
| GPIO11 | C18[1] | I/OL | — | DVDD3 | |
| GPIO12 | N18[1] | I/OL | — | DVDD3 | |
| GPIO13 | V11[1] | I/OL | — | DVDD3 | |
| GPIO14 | U11[1] | I/OL | — | DVDD3 | |
| GPIO15 | R18[1] | I/OL | — | DVDD3 | |
| GPIO16 | N17[1] | I/OL | — | DVDD3 | |
| GPIO17 | N16[1] | I/OL | — | DVDD3 | |
| **Digital Test** | | | | | |
| EJTAG_SEL | V7 | IL | — | DVDD3 | Tap controller select route:<br>■ 0 = JTAG pins to AR6001X TAP controller<br>■ 1 = JTAG pins to EJTAG TAP controller<br>Can be left open if not used. |
| TCK | V8 | IH | — | DVDD3 | JTAG clock input<br>Can be left open if not used. |
| TDI[2] | V9 | IH | — | DVDD3 | Test data input; default high<br>Can be left open if not used. |
| TDO[2] | U6 | OL | — | DVDD3_SDIO | Test data output; this bump must be pulled to the appropriate level during reset for interface configuration<br>Can be left open if not used. |
| TMS | U8 | IH | — | DVDD3 | Test mode select: default high<br>Can be left open if not used. |
| TRST_L | U7 | IL | — | DVDD3 | Test reset: active low, default low<br>Can be left open if not used. |

*Table 1-4.* **BGA Package Signal to Pin Relationships and Descriptions (continued)**

| Symbol | Pin | Type | Source or Destination | External PAD Power | Description |
|---|---|---|---|---|---|
| OE_L | M17 | I | Flash | — | Output enable for the flash part; must connect to MEM_OE_L on the board[3] |
| RESET_L | M2 | I | Flash | — | Reset to the flash part; must be tied to RST_OUT_L on the board[3] |
| RY/BY_L | M1 | I | Flash | — | Ready/Busy; not driven by the AR6001X, must be tied high |
| WE_L | L2 | I | Flash | — | Write enable for flash; must be connected to MEM_WE_L on the board[3] |
| **Memory** | | | | | |
| MEM_CS_0_L | V16 | OH | — | DVDD3 | Must be connected to CE_L |
| MEM_OE_L | L18 | OH | — | DVDD3 | Must be connected to OE_L |
| MEM_WE_L | N1 | OH | — | DVDD3 | Must be connected to WE_L |
| RST_OUT_L | N2 | OH | — | DVDD3 | The AR6001X asserts this pin when its core is in reset. Must be tied to RESET_L. |

[1]This pin is multiplexed. See Table 1-3, "BGA Package Multiplexed Pins," on page 11.
[2]TDI and TDO pins were swapped from version 1.1.
[3]This flash pin must be connected to a signal on the board, as noted.

| Symbol | Pin | Type | Voltage | Description |
|---|---|---|---|---|
| **Power** | | | | |
| AGND | A1, A10, B2, B10, C3, C9, C10, G7, G8, G9, G10, H7, H8, H9, H10, J2, J7, J8, J9, J10, K7, K8, K9, K10 | P | 0 V | Analog ground |
| AVDD18 | B1, C1, C2, C7, C8, C11, C12, F3, G3, H2, H3 | P | 1.8 V | Analog 1.8 V power supply. When LDO_BYPASS is enabled, these pins provide power to the analog core. |
| AVDD33 | A2, A3, A4, A12, B3, B4, B12, C4, C5, C13, D3, E3, J1 | P | 3.3 V | Analog 3.3 V power supply |
| DVDD18 | C14, C15, G16, H16, K2, L3, L16, M3, M16, N3, T7, T8, T11, T12 | P | 1.8 V | Digital 1.8 V power supply. When LDO_BYPASS is enabled, these pins provide power to the digital core. |
| DVDD18_SDIO | P1, P2 | P | 1.8 V | Digital power supply. When SDIO_LDO_BYPASS is enabled, these pins provide power to the SDIO block. |
| DVDD3 | C17, D16, D17, E16, E17, P3, P16, P17, R3, R16, R17, T4, T5, T14, T15 | P | 3.3 V | Digital 3.3 V power supply |
| DVDD3_ANT | B14, B15, B16 | P | | Digital power supply. Uses the same source as antenna control circuitry. |
| DVDD3_MEM | J17 | P | 3.3 V | Power supply for the stack flash memory |
| DVDD3_SDIO | U5, V5 | P | | Digital power supply for: CLK_REQ, GPIO_9, SDIO_CLK, SDIO_CMD, SDIO_DATA_0, SDIO_DATA_1, SDIO_DATA_2, SDIO_DATA_3, SDIO_LDO_BYPASS, SYS_RST_L |
| GND | A18, B17, C16, G11, G12, H11, H12, J3, J11, J12, J16, K3, K11, K12, K16, K17, L7, L8, L9, L10, L11, L12, M7, M8, M9, M10, M11, M12, T3, T9, T10, T16, U2, U9, U17, V1, V18 | P | 0 V | Ground |

| | | | | |
|---|---|---|---|---|
| **No Connection** | | | | |
| NC | F1, F2, G1, G2 | | — | Do not connect |
| RES | V6 | | — | Reserved, Internal pull-down. Can be left open or connected to ground. |

# 2. Functional Description

## 2.1 Overview

The AR6001X consists of integrated 802.11 MAC/BB/radio WLAN, MIPS R4kEm CPU core, SDIO, SPI, local bus, or memory bus interface, GPIOs for LED control, two-wire UART, digital audio $I^2S$ output for wireless speaker applications, serial interface that supports $I^2C$ and SPI, power and clock management for extended battery life. See "AR6011X ROCm Block Diagram" on page 1 and "Radio" on page 31.

## 2.2 LEDs

The AR6001X can drive LEDs using GPIO pins. An external NPN transistor can provide higher power drive. Note that the LED connects to the battery voltage. For multiple LED groups, multiple GPIOs can be assigned. The GPIO Sigma Delta PWM DAC can provide a continuous dimmer function. See "Sigma Delta DACs for GPIOs" on page 21.

## 2.3 Master SI/SPI Control

The AR6001X has a master serial interface (SI) that controls the EEPROM or other SPI devices. It can operate in two- or three-wire configurations. Software can add read/write and command/data signals as GPIO pins, and use serial interfaces in polling or interrupt mode. In polling mode, software should disable the interrupt in the interrupt controller block by clearing the associated interrupt enable bit.

A SI transaction consists of two phases: an optional data transmit phase of 0–8 bytes (see Table 2-1) followed by an optional data receive phase of 0–8 bytes (see Table 2-2).

The SI_CS register transmit (Tx) byte count field controls the size (in bytes) of the Tx phase, with a fixed source for each transmitted byte.

*Table 2-1.* **SI Transaction Tx Phase Bytes**

| Byte | Source |
|---|---|
| 0 | SI_TX_DATA0[7:0] |
| 1 | SI_TX_DATA0[15:8] |
| 2 | SI_TX_DATA0[23:16] |
| 3 | SI_TX_DATA0[31:24] |
| 4 | SI_TX_DATA1[7:0] |
| 5 | SI_TX_DATA1[15:8] |
| 6 | SI_TX_DATA1[23:16] |
| 7 | SI_TX_DATA1[31:24] |

The SI_CS register receive (Rx) byte count field controls the size (in bytes) of the Rx phase, with a fixed source for each received byte.

*Table 2-2.* **SI Transaction Rx Phase Bytes**

| Byte | Source |
|---|---|
| 0 | SI_RX_DATA0[7:0] |
| 1 | SI_RX_DATA0[15:8] |
| 2 | SI_RX_DATA0[23:16] |
| 3 | SI_RX_DATA0[31:24] |
| 4 | SI_RX_DATA1[7:0] |
| 5 | SI_RX_DATA1[15:8] |
| 6 | SI_RX_DATA1[23:16] |
| 7 | SI_RX_DATA1[31:24] |

To initialize the serial interface:

1. Write values to the SI_CLOCK register.
2. Configure the GPIO pins used for read/write and command/data signals.
3. In interrupt mode, set the SI interrupt enable and level in the interrupt controller. In polling mode, disable the SI interrupt in the interrupt controller.

To perform an SI transaction:

1. Write values into the assigned GPIO register to assert read/write and command/data.
2. Write appropriate values into the SI_CS register Tx and Rx byte count fields.
3. Write a 1 to the SI transaction SI_CS register start bit (combine with step 2 if desired).
4. To poll:
   a. Poll the transaction done indication bit in the SI_CS register until set, indicating the SI transaction is complete.
   b. When the Interrupt_Done bit is asserted, the CPU handles the interrupt and clears the interrupt_done bit. The next transaction can start with the write that clears the interrupt.
5. If the transaction includes an Rx phase, retrieve Rx data by reading bytes from the SI_RX_DATA0 and SI_RX_DATA1 registers.

## 2.4 GPIO

The AR6001X provides GPIO pins with direct CPU access. These pins can map independently to normal or fast CPU interrupts. Each pin has a GPIO_PIN register that configures its operation. Software should program each pin appropriately depending on the system design.

### 2.4.1 GPIO Pins

Table 2-3 describes GPIO_PIN fields.

*Table 2-3.* **GPIO_PIN Fields**

| Field | Description |
|---|---|
| GPIO_PIN WAKEUP_ENABLE | Software should set this bit if the GPIO pin indicates an event requiring immediate action, even in sleep mode. When set, the RTC power control FSM receives an interrupt caused by this pin so that interrupt events on this pin wake the chip. Wakeup_Enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup_Enable interrupts must be either rising edge or falling edge triggered. |
| GPIO_PIN INT_TYPE | Software can program the pin transition type, triggering an interrupt on either edge. Once the interrupt is set, it propagates to the interrupt controller on a signal dedicated to the GPIO, allowing software to assign different interrupt levels for different pins. |
| GPIO PAD_PULL and PAD_DRIVER | These fields allow software to control the pad type presented to the system. GPIO pads can be pull up, pull down, or no pull inputs. Outputs can be open drain or push/pull.<br>■ If the output is open drain, programming a “1” drives in high Z, a “0” drives to GND.<br>■ If the output is push/pull, “1” drives VDD, “0” drives to GND. |
| GPIO SOURCE | Each GPIO pin can drive data out from the GPIO_OUT register or from the Sigma Delta PWM. Multiple GPIOs can be programmed to output data from the same PWM resource. |

### 2.4.2 Reading and Writing the GPIO pins

After GPIO pins are configured with these registers, software can drive and sample the pins during normal operation. Table 2-4 describes the registers that provide mechanisms to drive and sample GPIO pins.

*Table 2-4.* **GPIO_PIN Fields**

| Field | Description |
|---|---|
| GPIO_OUT | This register drives all GPIO pins that use this register as their source. Software can write directly to the GPIO_OUT register and change the state of all GPIO pin outputs, or it can change only selected bits by writing to the GPIO_OUT_W1TS and GPIO_OUT_W1TC registers. These aliases to GPIO_OUT allow software to change only selected bits to a 0 or 1 without affecting other bits. For example, if a software driver only controls three GPIO pins, it changes the pins using GPIO_OUT_W1TS and GPIO_OUT_W1TC to avoid interference with other software drivers attached to a different set of GPIO pins. |
| GPIO_ENABLE | This register enables output drivers for individual GPIO pins. When the output driver is enabled, the corresponding GPIO_DATA_OUT bit or selected Sigma Delta PWM output value drives to the pin. When the output driver is not enabled, the pin is not driven and the GPIO pin becomes high Z. When a GPIO pin is used as an input only, the corresponding bit in this register should never be set. When a GPIO pin is used for input and output, the corresponding bit in this register should only be set when the pin needs to drive the output. This register also has aliases to set and clear only select bits. The GPIO_ENABLE_W1TC and GPIO_ENABLE_W1TS should be used by software drivers that only control a subset of the GPIO pins. |
| GPIO_STATUS | GPIO pins can cause interrupts when enabled using their INT_TYPE register. When a GPIO interrupt occurs, it sets the corresponding bit in the GPIO_STATUS register. All GPIO interrupt status bits are sent to the interrupt controller block by hardware, which assigns them different interrupt levels. Once a GPIO interrupt is handled, software should clear the corresponding bit in the GPIO_STATUS register. The software handler should use the GPIO_STATUS_W1TC alias to clear a single bit in an atomic fashion. |

### 2.4.3 Sigma Delta DACs for GPIOs

GPIO pins can also be driven from one of three Sigma Delta Pulse Waveform Modulator (PWM) DACs. The PWM resources can drive variable voltage outputs, including voltage controlled oscillators and LED arrays. For LED arrays, the PWMs allow continuous linear dimming depending on the value programmed in the target register. Software writes to the SIGMA_DELTA register to program a PWM resource. Set the ENABLE field to 1 when the resource is in use, or to 0 if the PWM is not needed to save power. The PRESCALAR field allows software to slow the PWM circuit by dividing its source clock. Depending on the output device, a slower or faster PWM output may be desirable. The PWM TARGET value represents the average number of pulses the output asserts over a 256 sample time period. For example, if TARGET is set to 128, the GPIO output pin asserts an average of half of the time. The Sigma Delta PWM can act for static or dynamic TARGET values. For dynamic values, the target value can change a maximum frequency of 256 * PRESCALAR for accurate DAC results.

## 2.5 UART

The AR6001X includes a high-speed Universal Asynchronous Receiver/Transmitter (UART) interface that is fully compatible with the 16550 UART industry standard. The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a modem, and parallel-to-serial conversion on data characters received from the CPU. The CPU can read UART status at any time during functional operation. Status information reported includes the type and condition of transfer operations performed by the UART as well as any error conditions. The UART includes a programmable data rate generator capable of dividing the timing reference clock input and producing a clock for driving internal transmitter logic. Provisions are also included to use this clock to drive the receiver logic. The UART has complete modem control capability, and a processor interrupt system.

The UART supports:

- Polling and interrupt modes
- Full duplex buffer system with 16-byte Tx/Rx FIFOs
- 5-, 6-, 7-, or 8-bit characters
- 1-, 1 1/2-, or 2-stop bit generation
- Odd, even, or no parity.
- Data rates of:
  - 230400 bps
  - 115200 bps
  - 57600 bps
  - 38400 bps
  - 28800 bps
  - 19200 bps
  - 9600 bps
  - 4800 bps
  - 2400 bps

## 2.6 Reset

All AR6001X reset control logic resides in the RTC block to ensure stable reset generation as long as a sufficient battery voltage is provided.

### 2.6.1 COLD_RESET

Cold reset is a hard reset that clears all chip states. It can be driven from the SYS_RST_L pin or a CPU write to the RTC_RESET register.

### 2.6.2 WARM_RESET

Warm reset is a partial hard reset that resets the chip but retains certain states. It is driven from the RTC_RESET register.

### 2.6.3 Reset Sequence

After COLD_RESET, the AR6001X enters WLAN_OFF to await an enable event from the ROC host. Its embedded CPU does not execute until the external host enables it. A typical AR6001X COLD_RESET sequence consists of:

1. The host system de-asserts CHIP_PWD, if asserted (CHIP_PWD assertion is optional, but de-assertion is necessary to use the AR6001X). The SYS_RST_L pin must be asserted for a short period of time (1 ms is sufficient) to start the AR6001X.
2. The host system de-asserts SYS_RST_L. The AR6001X latches the input level on GPIO9 and TDO to determine the host interface. See Table 1-2 on page 11.
3. For SDIO and SPI interface modes, the AR6001X enters the WLAN_OFF state. The host then reads interface registers to determine the function type the AR6001X supports. In local bus mode, skip this step and proceed to step 4
4. When ready for the WLAN, the host writes to the function enable bit to enable the AR6001X.

The AR6001X enters WAKEUP then ROC_ON and enables the MIPS CPU to begin booting. Firmware configures AR6001X functions and interfaces. When the AR6001X is ready to receive commands from the host, it sets the function ready bit.

## 2.7 Power Management

The AR6001X provides integrated power management and control functions and extremely low power operation for maximum battery life across all operational states by:

- Gating clocks for logic when not needed
- Shutting down unneeded high speed clock sources
- Reducing voltage levels to specific blocks in some states
- Reducing Tx and Rx active duty cycles
- Lowering CPU frequency when computational load is reduced

### 2.7.1 Hardware Power States

AR6001X hardware has six top level hardware power states managed by the RTC block. Table 2-5 describes the input from the MAC, CPU, SDIO/MBOX, interrupt logic, and timers effect the power states. Figure 2-1 depicts the state transition diagram.

*Table 2-5.* **AR6001X Hardware States**

| State | Description |
|---|---|
| OFF | ■ CHIP_PWD pin assertion immediately brings the chip to this state<br>■ LF_XTAL (32.678 KHz sleep clock) is disabled<br>■ LDOs are all off<br>■ No state is preserved<br>■ When CHIP_PWD de-asserts, the system must assert SYS_RST_L until the power has stabilized |
| WLAN _OFF | ■ WLAN is turned off<br>■ LF_XTAL (32.678 KHz sleep clock) is disabled<br>■ SDIO interface is on<br>■ Once the host enables the SDIO with a CCCR register write, the system begins to boot<br>■ Embedded CPU and MAC do not retain state |
| SLEEP | ■ Only the 32.768 KHz sleep clock is operating<br>■ The high speed crystal or oscillator is disabled for deep sleep<br>■ The digital core block is powered on, but its clocks are gated off at the clkmod block<br>■ All internal states are maintained |
| WAKEUP | ■ The system transitions from deep sleep to On.<br>■ The high frequency clocks gate off as the crystal or oscillator is brought up and the PLL enabled<br>■ Wakeup duration is programmable (default 3.8 ms); the wakeup state is bypassed for light sleep |
| ROC_ON | ■ The high speed clock is operational and sent to each block<br>■ The MAC is asleep, and MAC clocks are gated off<br>■ CPU, memory, MBOX, SDIO, and peripheral blocks are all operational<br>■ The CPU may be in WAIT state |
| ON | ■ The high speed clock is operational and sent to each block enabled by the clock control register<br>■ Lower level clock gating is implemented at the block level, including the CPU, which can be gated off from WAIT instruction while the system is on |

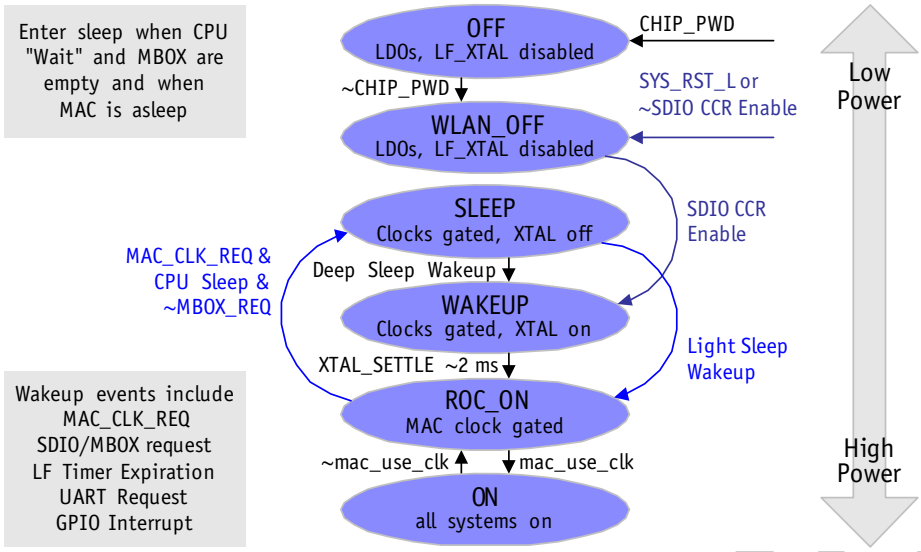



*Figure 2-1.* **AR6001X State Transitions**

### *2.7.2 Sleep State Management*

Sleep state minimizes power consumption while saving system states. In deep sleep state, all high speed clocks are gated off and the external crystal is powered off. Light sleep is similar to deep sleep, but the XTAL remains running for faster wakeup. For the AR6001X to enter sleep state, the MAC, SDIO/MBOX, and CPU systems must be in sleep state.

When the embedded MIPS CPU executes the WAIT command, the SDIO/MBOX is idle and the MAC system is in sleep state, the AR6001X enters the system Sleep state. In sleep state, the system gates all clock trees based on REF_CLK with only the sleep clock logic operating. The system remains in sleep state until a wakeup event causes the system to enter wakeup state, wait for the high frequency clock source to stabilize, and finally ungate all enabled clock trees. The CPU exits the WAIT state only when an interrupt arrives, which may result from the system wakeup event.

## *2.8 System Clocking*

Figure 2-2 describes the AR6001X clock control.

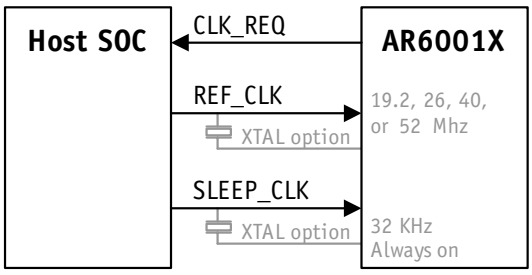



*Figure 2-2.* **AR6001X Clock Control**

### *2.8.1 SLEEP_CLK*

The AR6001X sleep clock must run at all times while the AR6001X is powered on. The sleep clock is a 32.768 KHz clock sourced from either an external crystal or from an external oscillator source. The AR6001X crystal interface detects the clock source type and enables the on-chip crystal driver if appropriate.

### *2.8.2 REF_CLK*

REF_CLK is the primary clock source for the analog and digital systems. It is a 19.2, 26, 40, or 52 MHz clock sourced from either an external crystal or oscillator source. It is the input to the RF synthesizer for generating required frequencies for proper 802.11 operation. An on-chip PLL creates the appropriate clock frequency for digital logic. When the AR6001X is in SLEEP state, REF_CLK is not needed. To minimize power consumption, the REF_CLK generator shuts down during deep sleep. If an external crystal is being used, the AR6001X disables the on-chip oscillator driver. If REF_CLK is coming from an external oscillator source, the AR6001X de-asserts its CLK_REQ signal and the external clock source may shut down REF_CLK.

When the AR6001X exits SLEEP state, it enters WAKEUP state and asserts CLK_REQ or enables its internal crystal oscillator depending on the clock configuration. The AR6001X remains in WAKEUP state for a programmable duration that must cover clock settling time. CLK_REQ remains asserted in WAKEUP and ON states.

### 2.8.3 Interface Clock

In some interface modes, the AR6001X can also receive a clock from SDIO or SPI interface logic during interface transactions. This clock drives interface logic and some registers accessible by the host in this clock domain, which allows the host to probe some AR6001X information, including SDIO/SPI Common I/O Area (CIA) when the AR6001X is in SLEEP state.

### 2.8.4 Clock Generators

The AR6001X derives all digital clocks from the REF_CLK reference clock with the exception of a small amount of logic driven by SLEEP_CLK.

- CORE clock
  AHB, APB, memory controller, interrupt controller, and DMA engine all run off the CORE_CLK. This clock's frequency is programmable to PLL output divisors.
- CPU Clock
  The CPU_CLK drives the MIPS CPU. This clock's frequency is programmable to PLL output divisors.
- MAC Clock
  MAC_CLK drives the MAC and baseband logic. This clock runs at a fixed frequency of 40 or 44 MHz depending on WLAN mode.
  The MAC asserts MAC_CLK_REQ when MAC_CLK is required. The asserting edge of MAC_CLK_REQ is followed by a delay of XTAL_SETTLE cycles of SLEEP_CLK before the MAC_CLK is available.
- SDIO_CLK
  SDIO_CLK operates at the external SDIO frequency. Multiple frequency options can provide the system flexibility to manage power consumption in finer granularity.

### 2.8.5 Antenna Switching

The switch table (see Table 2-6) contains 12 entries, each 6 bits wide, and is indexed by:

- The antenna selected by the MAC.
- The state of the transceiver (idle, receive, or transmit).
- Controls for Rx attenuation.

When fast-receive antenna diversity is enabled, the baseband will temporarily override the antenna selected by the MAC once a packet has been detected.

Table 2-6 also shows location of the registers.

*Table 2-6.* **Switch Table**

| Chip State | Ant Select | Rx Atten | Register Location (address and bits) | Register Name |
|---|---|---|---|---|
| idle | — | — | 0x9910, bits [9:4] | BB_ANTENNA_CONTROL |
| Bluetooth active | — | — | 0x9910, bits [15:10] | BB_ANTENNA_CONTROL |
| Tx | 1 | — | 0x9960, bits [5:0] | BB_SWITCH_TABLE1 |
| Rx | 1 | no | 0x9960, bits [11:6] | |
| Rx | 1 | no | 0x9960, bits [17:12] | |
| Rx | 1 | yes | 0x9960, bits [23:18] (unused) | |
| Rx | 1 | yes | 0x9960, bits [29:24] | |
| Tx | 2 | — | 0x9964, bits [5:0] | BB_SWITCH_TABLE2 |
| Rx | 2 | no | 0x9964, bits [11:6] | |
| Rx | 2 | no | 0x9964, bits [17:12] | |
| Rx | 2 | yes | 0x9964, bits [23:18] (unused) | |
| Rx | 2 | yes | 0x9964, bits [29:24] | |

Each 6-bit register controls the following AR6001X outputs (listed in the order of the most significant bit to the least significant bit):

- Internal Rx LNA
- ANTE
- ANTD
- ANTC
- ANTB
- ANTA

The most significant bit of the register controls the internal Rx LNA. The least significant bit of the register is ANTA. ATNE, ANTD, ANTC, ANTB, and ANTA are general purpose outputs that can be used to control antenna selection and external LNA, for example. The actual signals used are application-specific (refer to the *AR60xx Reference Design Schematics* for implementation of the antenna control signals).

Reset will cause the AR6001X to enter the idle state. Bits [9:4] of BB_ANTENNA_CONTROL are reset to all zeros, and will be applied to the six outputs of the switch table.

Register BB_ANTENNA_CONTROL contains bits other than [9:4]. Therefore, unless the other bits are known from initialization, it is recommended that bits [9:4] be altered with a read-modify-write cycle.

# 3. Host Interfaces

## 3.1 SDIO/SPI Slave Interface

The AR6001X SDIO/SPI slave interface is compliant with SDIO version 1.1. The AR6001X can work in various modes of IO host configuration, including SDIO, SPI, and 16-bit Local Bus mode.

Table 3-1 shows pin settings for mode configuration, sampled during reset.

*Table 3-1.* **Pin Settings for Mode Configuration**

| GPIO9 | TDO | Configuration |
|---|---|---|
| 0 | 0 | Generic SPI Mode |
| 0 | 1 | SDIO Mode (Default, GPIO9 pin has weak internal pull down, but TDO pin must be pulled high on the board) |
| 1 | 0 | Local Bus Mode |
| 1 | 1 | Reserved |

## 3.2 SDIO Address Map

The AR6001X supports the SDIO Common Information Area (CIA) registers to identify and initialize the AR6001X. This includes the Card Common Control Register (CCCR) and Function Basic Register (FBR). The AR6001X also supports the CIS tuple space for CIS0 and CIS1. Apart from these base SDIO registers, all AR6001X interface communication works over the SDIO Function 1 address space.

## 3.3 SPI Interface

The SDIO interface can be configured as SPI, as defined in the SDIO interface. SPI mode uses only one data line and contains a dedicated interrupt line. On the functional, register, and addressing levels, SPI is identical to SDIO. See the SDIO interface for details on physical SPI signalling.

## 3.4 Local Bus Interface

The AR6001X Local Bus slave interface can be configured to communicate with many host interface designs, including CardBus variants.

## 3.5 Host Interface Address Map

The host sees the same address map interface regardless of the physical interface used, thus allowing the software layer above the physical interface to be identical across physical interface types. The Local Bus interface supports 11 address bits so the lower 2 KB of address space must map all interface registers. The SDIO/SPI interface can use mailbox aliases above 2 KB as these aliases provide larger window interfaces for increased performance.

Figure 3-1 shows the host interface address map.

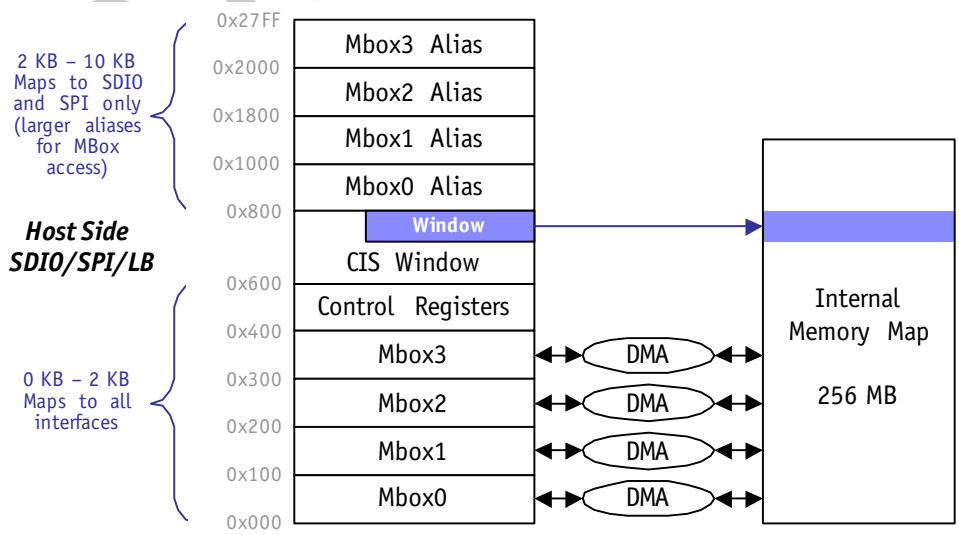



*Figure 3-1.* **Host Interface Address Map**

## 3.6 Mailboxes

The AR6001X supports four full duplex mailboxes to move messages between the AR6001X and the external host. Messages include packets, control messages, or any software-defined communication. AR6001X hardware use End of Message (EOM) markers to denote the end of a message that spans one or more memory descriptors on the AR6001X side.

Flow control of the four mailboxes must be managed by software. To assist software flow control, hardware provides eight counters as a credit mechanism. The counters may count messages, memory buffers, packets, or any unit that software defines. The host and AR6001X CPUs can read and write these counters using ordinary writes or atomic operations. Counter resource use is optional.

### 3.6.6 Error Conditions

If the host driver and AR6001X software lose flow control synchronization for any reason, mailbox errors conditions could arise.

- Tx Mailbox Overflow

  If no DMA descriptors are available on the AR6001X Tx side but the host still sends a message, the Tx Mailbox stalls the host physical interface. If the host interface remains stalled with the Tx FIFO full for a timeout period FIFO_TIMEOUT, a timeout error occurs. An interrupt is sent to the AR6001X CPU and the Host CPU. If the host status overflow bit is set, any mailbox Tx bytes that arrive from the host when the mailbox is full are discarded. When the host clears overflow interrupt, mailbox FIFOs return to normal operation. Software must then either resynchronize flow control state or reset the AR6001X to recover.

- Rx Mailbox Underflow

  If the host DMA engine reads a mailbox that does not contain any data, the host physical interface stalls. If this condition persists for more than a timeout period, the host and the AR6001X are sent an underflow error interrupt. As long as the host status underflow bit is set, any mailbox reads that arrive when the mailbox is empty return garbage data. When the host clears underflow interrupt, mailbox FIFOs return to normal operation. Software must then either resynchronize flow control state or reset the AR6001X to recover.

## 3.7 Interrupts

This section summarizes how interrupts flow between the AR6001X CPU and Host CPU. All interrupts can be masked by control registers.

### 3.7.1 AR6001X to Host

- The AR6001X CPU writes to the CPU_INT_STATUS register
- Data ready

  Rx FIFO is not empty (clears on Rx FIFO empty)
- Error interrupts, underflow or overflow
- Wake up interrupt

  Set when the AR6001X exits sleep
- Flow control

  Any COUNT goes from 0 to 1 (cleared when COUNT goes 1 to 0)
- Option

  All AR6001X internal interrupts can be mapped to the host in case the host wants to take complete control of the AR6001X MAC and resources

### 3.7.2 Host to AR6001X

- Host writes to INT_WLAN
- Error Interrupts (underflow or overflow)
- TX_CNT goes from 1 to 0 (out of descriptors)

### 3.7.3 SDIO Interface

The AR6001X interface is compliant with SDIO v1.1 and supports the SDIO common information area (CIA) registers for identifying and initializing the AR6001X. These registers include the card common control register (CCCR) and function basic register (FBR) as well as CIS tuple space for CIS0 and CIS1. All other interface communication occurs in SDIO function 1 address space. Figure 3-2 shows the generic SDIO address map.

The SDIO interface can be configured as SPI as defined in the SDIO interface. SPI mode uses only one data line and has a dedicated interrupt line. On the functional, register, and addressing level, SPI is identical to SDIO. See the SDIO interface for details on physical SPI signalling.

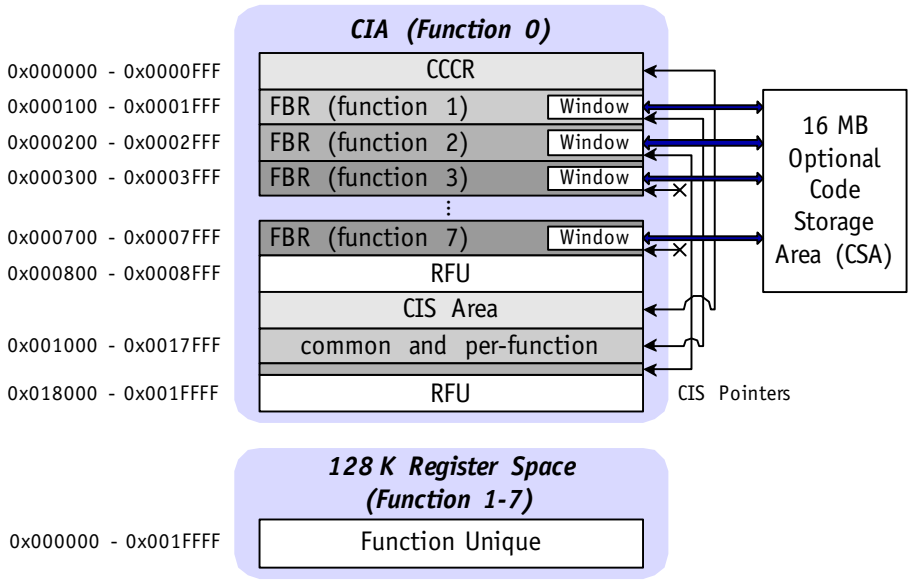



*Figure 3-2.* **Generic SDIO Address Map**

### *3.7.4 Local Bus Interface*

The AR6001X Local Bus interface is compatible with PCMCIA (PC Card) specification Rev. 7.0 and with standard SRAM/Flash memory interface (Local Bus). It includes a bootstrap option that must be applied during SYS_RST_L de-assertion.

| Configuration of Local Bus Interface | |
|---|---|
| PC Card | Pin LB_WAIT_L is pulled UP at SYS_RST_L de-assertion. The LB_BE_L[1:0] pins are used as per-byte chip-selects. The LB_CS_L pin is not used. |
| Local Bus | Pin LB_WAIT_L is pulled DOWN at SYS_RST_L de-assertion. The LB_CS_L is used as a word chip select and the LB_BE_L[1:0] pins are used as byte enables. |

To use the SDIO interface at 1.8 V:

1. Assert SDIO_LDO_BYPASS
2. Supply 1.8 V to DVDD3_SDIO and DVDD18_SDIO

# 4. Radio

The AR6001X transceiver consists of four major functional blocks (see Figure 4-1):

- Receiver (RX)
- Transmitter (TX)
- Frequency synthesizer (SYNTH)
- Associated bias/control (BIAS)

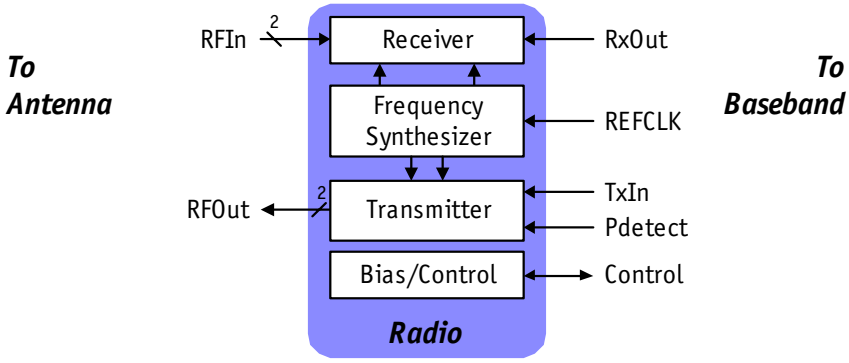



*Figure 4-1.* **Radio Functional Block Diagram**

## *4.1 Receiver (Rx) Block*

The receiver converts an RF signal (with 20 MHz or 40 MHz bandwidth) to baseband I and Q outputs. The input frequency range of the receiver is 2.4 GHz for IEEE 802.11b and 802.11g signals and 4.9 to 5.925 GHz for IEEE 802.11a signals.

The receiver implements an integrated down-conversion architecture that eliminates the requirement for an external intermediate frequency filter while providing the advantages of traditional heterodyne approaches. The receiver topology includes a low noise amplifier (LNA), a radio frequency (RF) mixer, an intermediate frequency (IF) mixer, and a baseband programmable gain amplifier (PGA) as shown in Figure 4-2. The RF mixer converts the output of the on-chip LNA to an intermediate frequency. The IF mixer converts this signal down to baseband I and Q signals. The I and Q signals are low-pass filtered and amplified by a baseband programmable gain filter controlled by digital logic. The baseband I and Q signals are sent to the ADC.

The DC offset of the receive chain is reduced using multiple DACs controlled by the MAC/Baseband block. Additionally, the receive chain can be digitally powered down to conserve power.



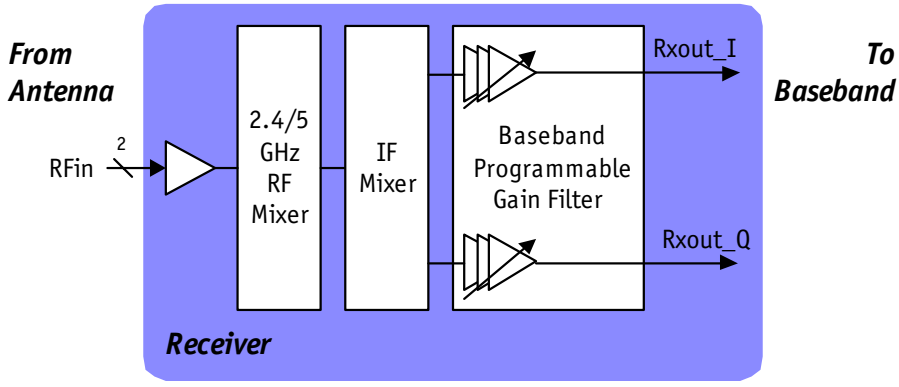



*Figure 4-2.* **Radio Receiver Block Diagram**

## 4.2 Transmitter (TX) Block

The transmitter converts baseband I and Q inputs to 2.4 and 5 GHz RF outputs (see Figure 4-3).

AR6001X transmitter inputs are current outputs of the DAC. These currents are low-pass filtered through an on-chip reconstruction filter to remove spectral images and out-of-band quantization noise.

I and Q signals convert to RF signals using integrated up-conversion architecture.

The intermediate frequency (IF) mixer converts baseband signals to an intermediate frequency. The RF mixer converts IF signals into radio frequency signals, which are driven off-chip through a power amplifier.

The transmit chain can be digitally powered down to conserve power. To ensure that FCC limits are observed and output power stays close to the maximum allowed, transmit output power is adjusted by a closed loop digitally programmed control loop at the start of each packet. The closed-loop power control can be based on an on-chip or off-chip power detector. Refer to the *External Power Control for Design* application note for more details.

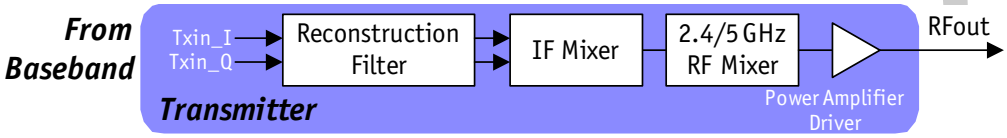



*Figure 4-3.* **Radio Transmitter Block Diagram**

### 4.2.1 Synthesizer (SYNTH) Block

The radio supports two on-chip synthesizers to generate local oscillator (LO) frequencies for receiver and transmitter mixers. Both synthesizers share the topology shown in Figure 4-4.

A 40-MHz crystal generates a signal used as the synthesizer reference input. An on-chip voltage controlled oscillator (VCO) provides the desired LO signal based on a phase/frequency locked loop.

The loop filter components are all integrated on-chip and can be digitally optimized through the serial interface. On power up or channel reselection, the synthesizer takes about 0.2 ms to settle.

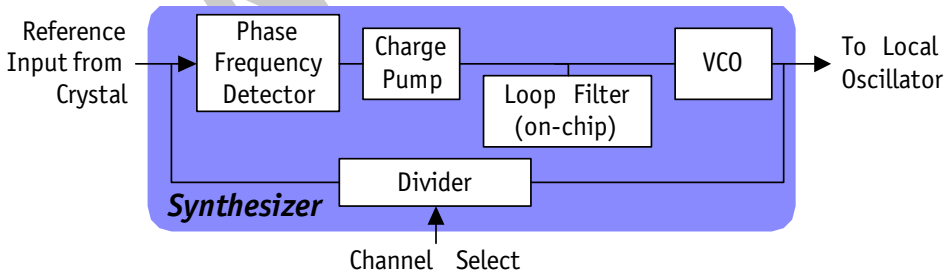



*Figure 4-4.* **Radio Synthesizer Block Diagram**

## 4.3 Bias/Control (BIAS) Block

The bias/control block provides reference voltages and currents for all other circuit blocks (see Figure 4-5). An on-chip bandgap reference circuit provides the needed voltage and current references based on an external 6.19 kΩ ± 1% resistor.

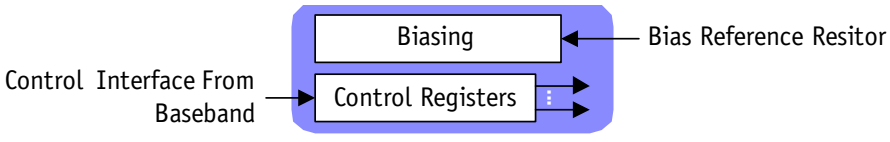



*Figure 4-5.* **Bias/Control Block Diagram**

# 5. Electrical Characteristics

## 5.1 Absolute Maximum Ratings

Table 5-1 summarizes the absolute maximum ratings and Table 5-2 lists the recommended operating conditions for the AR6001X. Absolute maximum ratings are those values beyond which damage to the device can occur. Functional operation under these conditions, or at any other condition beyond those indicated in the operational sections of this document, is not recommended.

**NOTE:** Maximum rating for signals follows the supply domain of the signals (e.g., the LDO_BYPASS maximum rating is –0.3 V to 4.0 V).

*Table 5-1.* **Absolute Maximum Ratings**

| Symbol | Parameter | Max Rating | Unit |
|---|---|---|---|
| $V_{dd18}$ | 1.8 V supply voltage | –0.3 to 2.5 | V |
| $V_{dd33}$ | 3.3 V supply voltage | –0.3 to 4.0 | V |
| $RF_{in}$ | Maximum RF input (reference to 50 Ω) | +10 | dBm |
| $T_{store3}$ | Storage temperature | –45 to 135 | °C |
| ESD | Electrostatic discharge tolerance | 2000 | V |

## 5.2 Recommended Operating Conditions

*Table 5-2.* **Recommended Operating Conditions**

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{dd18}$ | Supply voltage | — | 1.71 | 1.8 | 1.89 | V |
| $V_{dd33}$ | Supply voltage | — | 2.9 | 3.30 | 3.6 | V |
| $T_{ambient}$ | Ambient temperature | — | –40 | 25 | 85 | °C |

## 5.3 DC Electrical Characteristics

Table 5-3 and Table 5-4 list the general DC electrical characteristics.

The conditions in Table 5-3 apply to all DC characteristics unless otherwise specified:

$V_{dd}$ = 3.3 V, $T_{amb}$ = 25 °C

*Table 5-3.* **General DC Electrical Characteristics (Vdd = 3.3 V)**

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{IH}$ | High Level Input Voltage | | 2.0 | — | $V_{dd}$ + 0.3 | V |
| $V_{IL}$ | Low Level Input Voltage | | –0.3 | — | 0.8 | V |
| $I_{IL}$ | Input Leakage Current | Without Pull-up or Pull-down | — | ± 5 | — | µA |
| | | With Pull-up or Pull-down | — | ± 65 | — | µA |
| $V_{OH}$ | High Level Output Voltage | No Load ($I_o$ = 0) | $V_{dd}$ –0.3 | — | — | V |
| | | $I_o$ = 12 mA | $V_{dd}$ –0.8 | — | — | V |
| $V_{OL}$ | Low Level Output Voltage | No Load ($I_o$ = 0) | — | — | 0.20 | V |
| | | $I_o$ = 12 mA | — | — | 0.27 | V |
| $I_O$ | Output Current (SYS_RESET_L, LCL_0, LCL_1, LCL_2, LCL_3, GPIO_0, GPIO_1, GPIO_2, GPIO_3, M1_MDIO, MEM_WE_L, SD_CS_L, SD_CLK) | $V_o$ = 0 to $V_{dd}$ | — | — | TBD | mA |
| $I_O$ | Output Current All other digital output pins | $V_o$ = 0 to $V_{dd}$ | — | — | 8 | mA |
| $C_{IN}$ | Input Capacitance | — | — | 6 | — | pF |
| R | Pull-up and Pull-down Resistance of Input Ports | — | — | 2 | — | MΩ |

The conditions in Table 5-4 apply to all DC characteristics unless otherwise specified:

$V_{dd}$ = 1.8 V, $T_{amb}$ = 25 °C

*Table 5-4.* **General DC Electrical Characteristics (Vdd = 1.8 V)**

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{IH}$ | High Level Input Voltage | | 1.2 | — | $V_{dd}$ + 0.3 | V |
| $V_{IL}$ | Low Level Input Voltage | | –0.3 | — | 0.5 | V |
| $I_{IL}$ | Input Leakage Current | Without Pull-up or Pull-down | — | ± 5 | — | µA |
| | | With Pull-up or Pull-down | — | ± 30 | — | µA |
| $V_{OH}$ | High Level Output Voltage | No Load ($I_o$ = 0) | $V_{dd}$ –0.3 | — | — | V |
| | | $I_o$ = 12 mA | $V_{dd}$ –0.4 | — | — | V |
| $V_{OL}$ | Low Level Output Voltage | No Load ($I_o$ = 0) | — | — | 0.20 | V |
| | | $I_o$ = 12 mA | — | — | 0.27 | V |
| $I_O$ | Output Current (SYS_RESET_L, LCL_0, LCL_1, LCL_2, LCL_3, GPIO_0, GPIO_1, GPIO_2, GPIO_3, M1_MDIO, MEM_WE_L, SD_CS_L, SD_CLK) | $V_o$ = 0 to $V_{dd}$ | — | — | TBD | mA |
| $I_O$ | Output Current All other digital output pins | $V_o$ = 0 to $V_{dd}$ | — | — | 8 | mA |
| $C_{IN}$ | Input Capacitance | — | — | 6 | — | pF |
| R | Pull-up and Pull-down Resistance of Input Ports | — | — | 2 | — | MΩ |

Figure 5-1 shows the power up/power down sequence for the AR6001X.

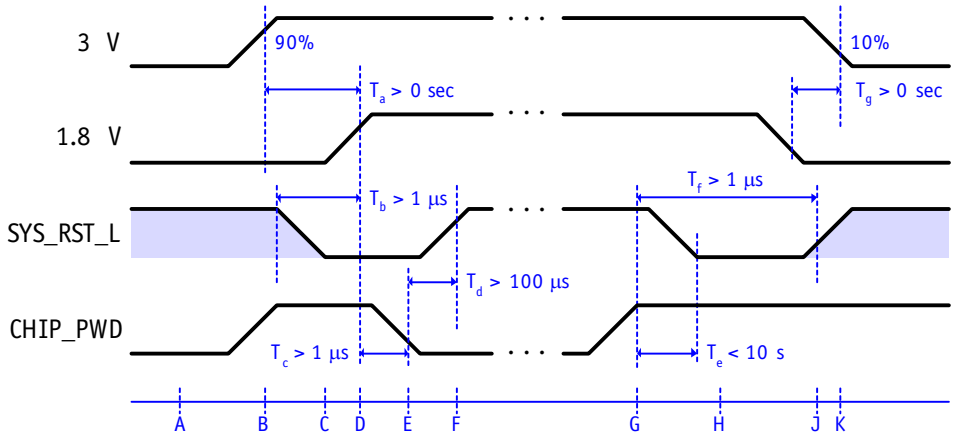



*Figure 5-1.* **Power Up/Power Down Sequence**

| | |
|---|---|
| A | Power on. SYS_RST_L may be high or low; 3 V, 1.8 V, CHIP_PWD are low |
| B | 3 V is valid (90%) and CHIP_PWD is asserted simultaneous to 3 V |
| C | SYS_RST_L asserts at least $T_b$ before 1.8 V is valid, placing the AR6001X in reset |
| D[1] | 1.8 V is valid at least $T_a$ after 3 V is valid. |
| E | CHIP_PWD de-asserts at least $T_c$ after 1.8 V is valid |
| F | SYS_RST_L de-asserts at least $T_d$ after CHIP_PWD de-asserts |
| G | Power down. SYS_RST_L asserts at most $T_e$ after CHIP_PWD asserts |
| H | CHIP_PWD asserts at least $T_f$ before 1.8 V is powered down |
| J | 1.8 V is invalid at least $T_g$ before 3 V is invalid |
| K | 3 V is invalid |

[1]$T_a$ may be substituted by the condition where the 1.8 V supply is always no more than 0.3 V higher than the 3 V supply.

## 5.4 Radio Receiver Characteristics

Table 5-5 summarizes the AR6001X receiver characteristics.

*Table 5-5.* **Receiver Characteristics for 2.4 GHz operation**

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $F_{rx}$ | Receive input frequency range | 5 MHz center frequency | 2.312 | — | 2.484 | GHz |
| NF | Receive chain noise figure | See Note [1] | — | 5.5 | — | dB |
| $S_{rf}$ | Sensitivity | See Note [2] | | | | dBm |
| | CCK, 1 Mbps | | — | –95 | — | |
| | CCK, 11 Mbps | | — | –90 | — | |
| | OFDM, 6 Mbps | | — | –92 | — | |
| | OFDM, 54 Mbps | | — | –73 | — | |
| IP1dB | Input 1 dB compression (min. gain) | — | — | –10 | — | dBm |
| IIP3 | Input third intercept point (min. gain) | — | — | –1 | — | dBm |
| $Z_{RFin\_input}$ | Single-ended input impedance | See Note [3] | — | 15–j35 | — | — |
| $ER_{phase}$ | I,Q phase error | | — | 1 | — | degree |
| ERamp | I,Q amplitude error | | — | 0.5 | — | dB |
| $R_{adj}$ | Adjacent channel rejection | 10 to 20 MHz [4] | | | | dB |
| | CCK | | 35 | — | — | |
| | OFDM, 6 Mbps | | 16 | 20 | — | |
| | OFDM, 54 Mbps | | –1 | 3 | — | |
| TRpowup | Time for power up (from synth on) | — | — | 1 | — | µs |

[1] For improved sensitivity performance, an external LNA may be used.
[2] Sensitivity performance based on the Atheros reference design, which includes RF filter, Tx/Rx antenna switch, and an external LNA.
[3] Refer to the *AR6001 ROCm Reference Guide* for information.
[4] Measured with AR6001X.

*Table 5-6.* **Receiver Characteristics for 5 GHz operation**

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $F_{rx}$ | Receive input frequency range | 5 MHz center frequency | 4.90 | — | 5.925 | GHz |
| NF | Receive chain noise figure (max. gain) | See Note [1] | — | 5.5 | | dB |
| $S_{rf}$ | Sensitivity | See Note [2] | | | | dBm |
| | 6 Mbps | | — | –92 | — | |
| | 54 Mbps | | — | –73 | — | |
| IP1dB | Input 1 dB compression (min. gain) | | –10 | –7 | — | dBm |
| IIP3 | Input third intercept point (min. gain) | | 2 | 5 | — | dBm |
| $Z_{RFin\_input}$ | Single-ended input impedance | 5.15–5.825 GHz differential [3] | — | 20+j35 | — | — |
| $ER_{phase}$ | I,Q phase error | | — | 3.5 | 5 | degree |
| ERamp | I,Q amplitude error | | — | 0.5 | 1 | dB |
| $R_{adj}$ | Adjacent channel rejection | 10 to 20 MHz [4] | | | | dB |
| | 6 Mbps | | 16 | 22 | — | |
| | 54 Mbps | | –1 | 5 | — | |
| $R_{alt}$ | Alternate channel rejection | 20 to 30 MHz [4] | | | — | dB |
| | 6 Mbps | | 32 | 37 | | |
| | 54 Mbps | | 15 | 20 | | |
| $BB_{atten}$ | Baseband filter attenuation | | | | | dB |
| | 20 MHz offset | | — | –21 | –17 | |
| | 40 MHz offset | | — | –46 | –40 | |
| $BB_{ripple}$ | Baseband filter passband ripple | — | — | 0.4 | 1 | dB |
| TRpowup | Time for power up (from synth on) | — | — | 1 | — | µs |

[1]Measured using the balun recommended by Atheros.
[2]Sensitivity performance is based on the Atheros reference design, which includes RF filter, Tx/Rx antenna switch, and an external LNA.
[3]Refer to the *Hardware Design Guide* for information.
[4]Measured with AR6001X.

## 5.5 Radio Transmitter Characteristics

Table 5-7 summarizes the transmitter characteristics for the AR6001X.

*Table 5-7.* **Transmitter Characteristics for 2.4 GHz operation**

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $F_{tx}$ | Transmit output frequency range | 5 MHz center frequency | 2.312 | — | 2.484 | GHz |
| $P_{out}$ | Mask Compliant CCK output power | See Note [1] | — | 0 | — | dBm |
| | EVM Compliant OFDM output power for 64 QAM | See Note [1] | | –4 | — | dBm |
| $SP_{gain}$ | PA gain step | See Note [2] | — | 0.5 | — | dB |
| $A_{pl}$ | Accuracy of power leveling loop | See Notes [3] [4] | — | ± 0.5 | — | dB |
| $Z_{RFout\_load}$ | Recommended PA differential load impedance | See Note [5] | — | 50–j40 | — | — |
| OP1dB | Output P1dB (max. gain) | 2.442 GHz | — | 6[6] | — | dBm |
| OIP3 | Output third order intercept point (max gain) | 2.442 GHz | — | 13[6] | — | dBm |
| SS | Sideband suppression | | — | –40 | — | dBc |
| RS | Synthesizer reference spur | — | — | –65 | | dBc |
| $Tx_{mask}$ | Transmit spectral mask | See Note [7] | | | | dBr |
| | CCK | | | | | |
| | At 11 MHz offset | | –30 | –35 | — | |
| | At 22 MHz offset | | –50 | –53 | — | |
| | OFDM | | | | | |
| | At 11 MHz offset | | –20 | –27 | — | |
| | At 20 MHz offset | | –28 | –38 | — | |
| | At 30MHz offset | | –40 | –52 | — | |
| TTpowup | Time for power up (from synth on) | — | — | 1.5 | — | µs |

[1] Measured using the balun recommended by Atheros under closed-loop power control.
[2] Guaranteed by design.
[3] Manufacturing calibration required.
[4] Not including tolerance of external power detector and its temperature variation.
[5] Refer to the design guide for information.
[6] Programmable
[7] Measured at the antenna connector port. Average conducted transmit power levels = 20 dBm (CCK), 19 dBm at 64 QAM (OFDM). System includes external PA.

*Table 5-8.* **Transmitter Characteristics for 5 GHz Operation**

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $F_{tx}$ | Transmit output frequency range | 20 MHz center frequency | 4.9 | — | 5.925 | GHz |
| $P_{out}$ | EVM Compliant OFDM output power for 64 QAM | See Note [1] | — | 5 | — | dBm |
| $SP_{gain}$ | PA gain step | See Note [2] | — | 0.5 | — | dB |
| $A_{pl}$ | Accuracy of power leveling loop | See Note [3] | — | ± 0.5 | ± 1.5 | dB |
| $Z_{RFout\_load}$ | Recommended PA differential load impedance | 5.15 – 5.825 GHz differential [4] | — | 180+ j150 | — | — |
| OP1dB | Output P1dB (max. gain) | 5.25 GHz | 6[5] | 8.5 | — | dBm |
| OIP3 | Output third order intercept point (max gain) | 5.25 GHz | 14.5[5] | 17.5 | — | dBm |
| SS | Sideband suppression | | — | –45 | –30 | dBc |
| $LO_{leak}$ | LO leakage: at 2/3 of the RF output | | | | | dBm |
| | @ RF=5.15-5.35 GHz (FCC) | | — | –65 | –60 | |
| | @ RF=5.35-5.725 GHz (ETSI) | | — | –70 | –65 | |
| | @ RF=5.725-5.825 GHz (FCC) | | — | –70 | –65 | |
| RS | Synthesizer reference spur | | — | –55 | — | dBc |
| $Tx_{mask}$ | Transmit spectral mask | See Note [6] | | | | dBr |
| | At 11 MHz offset | | –20 | –22 | — | |
| | At 20 MHz offset | | –28 | –32 | — | |
| | At 30 MHz offset | | –40 | –52 | — | |
| TTpowup | Time for power up (from synth on) | | — | 1.5 | — | µs |

[1]Measured using the balun recommended by Atheros under closed-loop power control. The use of an external PA with external power detector is recommended. See the application note *External Power Control for Design Using AR5002*.
[2]Guaranteed by design.
[3]Manufacturing calibration required.
[4]Refer to the design guide for information.
[5]Programmable
[6]Measured at the antenna connector port. Average conducted transmit power levels = 18 dBm at 64 QAM (OFDM). System includes external PA.

## 5.6 AR6001X Synthesizer Characteristics

Table 5-9 summarizes the synthesizer characteristics for the AR6001X.

*Table 5-9.* **Synthesizer Composite Characteristics for 2.4 GHz Operation**

| Symbol | Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| Pn | Phase noise (at Tx_Out) | | | | | dBc/Hz |
| | At 30 KHz offset | | — | –105 | — | |
| | At 100 KHz offset | | — | –105 | — | |
| | At 500 KHz offset | | — | –105 | — | |
| | At 1 MHz offset | | — | –120 | — | |
| $F_c$ | Center channel frequency | Center frequency at 5 MHz spacing [1] | 2.312 | — | 2.484 | GHz |
| $F_{ref}$ | Reference oscillator frequency | ± 20 ppm | — | 40 | — | MHz |
| $F_{step}$ | Frequency step size (at RF) | See Note | — | 1 | — | MHz |
| $TS_{powup}$ | Time for power up (from sleep) | — | — | 0.2 | — | ms |

[1]Frequency is measured at the TX output.

*Table 5-10.* **Synthesizer Composite Characteristics for 5 GHz Operation**

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Pn | Phase noise (at Tx_Out) | | | | | dBc/Hz |
| | At 30 KHz offset | | — | –100 | –95 | |
| | At 100 KHz offset | | — | –98 | –93 | |
| | At 500 KHz offset | | — | –105 | –100 | |
| | At 1 MHz offset | | — | –112 | –107 | |
| $F_c$ | Center channel frequency | Center frequency at 5 MHz spacing [1] | 4.90 | — | 5.925 | GHz |
| $F_{ref}$ | Reference oscillator frequency | ± 20 ppm | — | 40 | — | MHz |
| $F_{step}$ | Frequency step size (at RF) | See Note [2] | — | 5 | — | MHz |
| $TS_{powup}$ | Time for power up (from sleep) | — | — | 0.2 | — | ms |

[1]Frequency is measured at the Tx output.
[2]5 MHz channel spacing is for the 5.725 to 5.925 GHz band.

## 5.7 Power Consumption Parameters

The following conditions apply to the typical characteristics unless otherwise specified:

$V_{dd18}$ = 1.8 V
$V_{dd33}$ = 3.3 V, $T_{amb}$ = 25 °C

Table 5-11 shows the typical power drain on each of the on-chip power supply domains as a function of the AR6001X's operating mode.

*Table 5-11.* **Total 2.4 GHz System Power Including PA and LNA[1] with External 1.8V**

| 802.11b/g | 3.3 V Supply | 1.8 V Supply | Total | Unit |
|---|---|---|---|---|
| Sleep [2] | 120 | 180 | 300 | µW |
| Tx [3] | 380 | 260 | 640 | mW |
| Rx (max. gain)[4] | 152 | 280 | 432 | mW |

[1]Typical PA power (15 dBm) is 264 mW. Typical LNA power is 36 mW.
[2]Powered-down state; only the CLK40 pads and crystal oscillator are on.
[3]Transmitter and synthesizer are on.
[4]Receiver and synthesizer are on with maximum receive gain.

*Table 5-12.* **Total 5 GHz System Power Including PA and LNA[1] with External 1.8V**

| 802.11a | 3.3 V Supply | 1.8 V Supply | Total | Unit |
|---|---|---|---|---|
| Sleep [2] | 120 | 180 | 300 | µW |
| Tx [3] | 367 | 247 | 614 | mW |
| Rx (max. gain)[4] | 110 | 290 | 400 | mW |

[1]Typical PA power (15 dBm) is 264 mW. Typical LNA power is 36 mW.
[2]Powered-down state; only the CLK40 pads and crystal oscillator are on.
[3]Transmitter and synthesizer are on.
[4]Receiver and synthesizer are on with maximum receive gain.

*Table 5-13.* **Total 2.4 GHz System Power Including PA and LNA[1] with Internal 1.8V**

| 802.11b/g | 3.3 V Supply | Total | Unit |
|---|---|---|---|
| Sleep [2] | 660 | 660 | µW |
| Tx [3] | 856 | 856 | mW |
| Rx (max. gain)[4] | 665 | 665 | mW |

[1]Typical PA power (15 dBm) is 264 mW. Typical LNA power is 36 mW.
[2]Powered-down state; only the CLK40 pads and crystal oscillator are on.
[3]Transmitter and synthesizer are on.
[4]Receiver and synthesizer are on with maximum receive gain.

*Table 5-14.* **Total 5 GHz System Power Including PA and LNA[1] with Internal 1.8V**

| 802.11a | 3.3 V Supply | Total | Unit |
|---|---|---|---|
| Sleep [2] | 660 | 660 | µW |
| Tx [3] | 820 | 820 | mW |
| Rx (max. gain)[4] | 642 | 642 | mW |

[1]Typical PA power (15 dBm) is 264 mW. Typical LNA power is 36 mW.
[2]Powered-down state; only the CLK40 pads and crystal oscillator are on.
[3]Transmitter and synthesizer are on.
[4]Receiver and synthesizer are on with maximum receive gain.

# 6. AC Specifications

The AR6001X interface supports several physical host standards with only one host interface standard active at a time. The host interface type is determined at SYS_RST_L de-assertion.

The AR6001X's interface bus can be configured to be in SPI, SDIO, or local bus mode. Table 6-1 shows pin settings for mode configuration, sampled during reset.

*Table 6-1.* **Pin Settings for Mode Configuration**

| GPIO9 | TDO | Configuration |
|---|---|---|
| 0 | 0 | Generic SPI Mode |
| 0 | 1 | SDIO Mode (Default, GPIO9 pin has weak internal pull down, but TDO pin must be pulled high on the board) |
| 1 | 0 | Local Bus Mode |
| 1 | 1 | Reserved |

## 6.1 External 32 KHz Input Clock Timing

Figure 6-1 and Table 6-2 show the external 32 KHz input clock timing requirements.

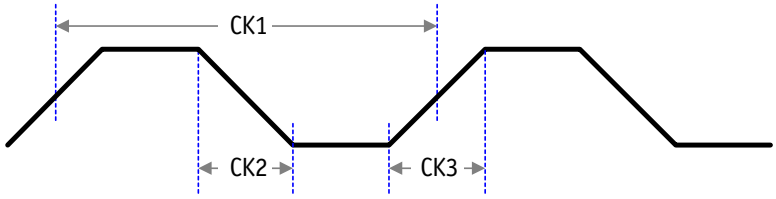



*Figure 6-1.* **External 32 KHz Input Clock Timing Requirements**



*Table 6-2.* **External 32 KHz Input Clock Timing[1]**

| Symbol | Description | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| CK1 | Frequency | — | 32.768 | — | KHz |
| CK2 | Fall time | — | — | 100 | ns |
| CK3 | Rise time | — | — | 100 | ns |
| CK4 | Duty cycle (high-to-low ratio) | 30 | — | 70 | % |
| CK5 | Frequency stability | –50 | — | 50 | ppm |
| CK6 | Input high voltage | $V_{dd18}$ – 0.6 | — | $V_{dd18}$ + 0.3 | V |
| CK7 | Input low voltage | –0.3 | — | 0.55 | V |

[1]These data assume 1.8 V voltage rails, VDD = 1.8 V.

## 6.2 Local Bus Interface Timing

Figure 6-2 shows the read timing for a local bus style transaction.

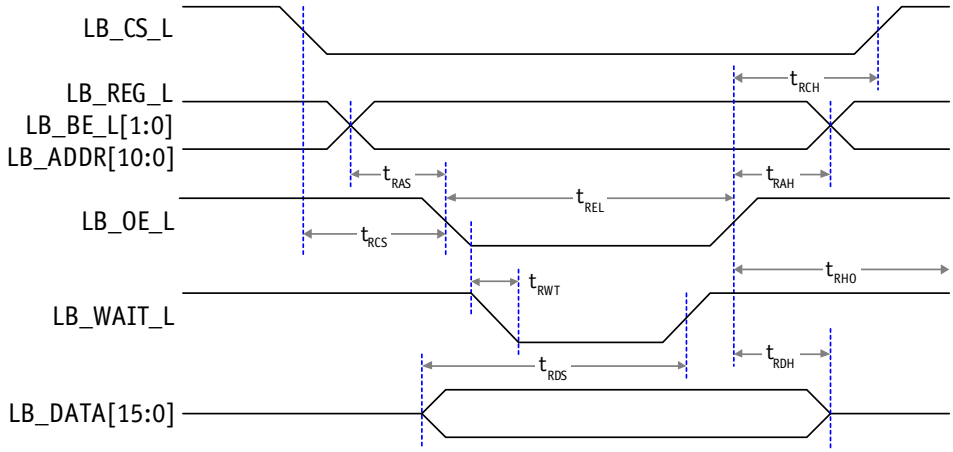



*Figure 6-2.* **Local Bus Style Read Interaction Timing**

Figure 6-3 shows the write timing for a local bus style transaction.

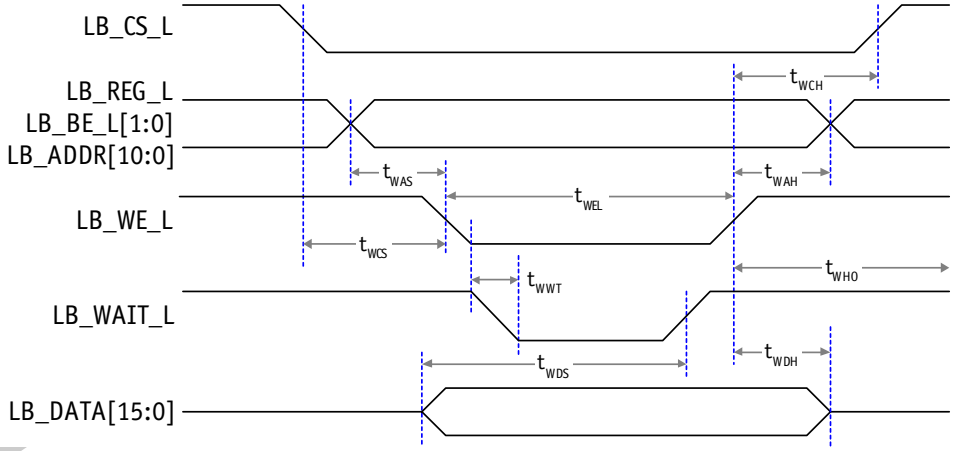



*Figure 6-3.* **Local Bus Style Write Interaction Timing**

Table 6-3 shows the initial pre-production values for timing constraints.

*Table 6-3.* **Timing Constraint Pre-Production Values**

| Timing Constraint | Value | Description |
|---|---|---|
| $t_{RCH}$, $t_{WCH}$ | 12 ns | Read/Write Chip Select Hold Time |
| $t_{RCS}$, $t_{WCS}$ | 12 ns | Read/Write Chip Select Setup Time |
| $t_{RAH}$, $t_{WAH}$ | 12 ns | Read/Write Address Hold Time |
| $t_{RAS}$, $t_{WAS}$ | 12 ns | Read/Write Address Setup Time |
| $t_{RHO}$, $t_{WHO}$ | 30 ns/80 ns | Read/Write Hold off (time before next LB_WE_L or LB_OE_L assertion) |
| $t_{RDH}$, $t_{WDH}$ | 50 ns/12 ns | Read/Write Data Hold Time |
| $t_{RDS}$, $t_{WDS}$ | 12 ns | Read/Write Data Setup Time |
| $t_{REL}$, $t_{WEL}$ | 30 ns | Read/Write Enable Length (minimum length of LB_WE_L or LB_OE_L pulse) |
| $t_{REL}$, $t_{WEL}$ (no LB_WAIT_L) | 200 ns | Read/Write Enable Length (minimum length of LB_WE_L or LB_OE_L pulse when not using LB_WAIT_L for flow control) |
| $t_{RWT}$, $t_{WWT}$ | 12 ns | Read/Write Wait Valid Time (maximum time between OE/WE and WAIT assertion) |

## 6.3 SD/SPI Interface Timing

Figure 6-4 shows the write timing for a SD/SPI style transaction.

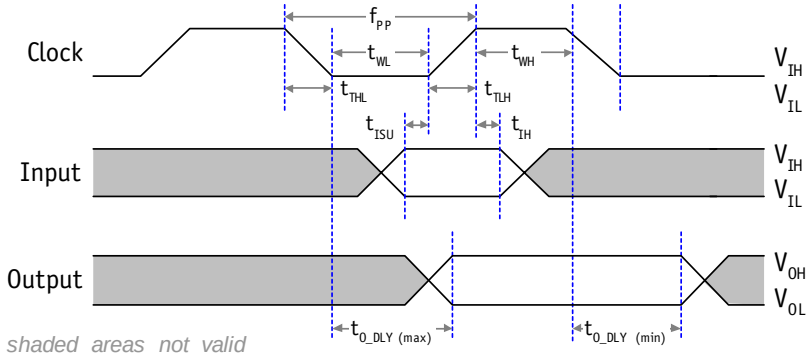



*Figure 6-4.* **SD/SPI Timing**

Table 6-4 shows the initial pre-production values for timing constraints.

*Table 6-4.* **SD/SPI Timing Constraints**

| Parameter | Description | Min | Max | Unit | Note |
|---|---|---|---|---|---|
| $f_{PP}$ | Clock frequency data transfer mode | 0 | 25 | MHz | 100 pF ≥ $C_L$ (7 cards) |
| $t_{WL}$ | Clock low time | 10 | — | ns | 100 pF ≥ $C_L$ (7 cards) |
| $t_{WH}$ | Clock high time | 10 | — | ns | 100 pF ≥ $C_L$ (7 cards) |
| $t_{TLH}$ | Clock rise time | — | 10 | ns | 100 pF ≥ $C_L$ (10 cards) |
| $t_{THL}$ | Clock fall time | — | 10 | ns | 100 pF ≥ $C_L$ (7 cards) |
| $t_{ISU}$ | Input setup time | 5 | — | ns | 25 pF ≥ $C_L$ (1 card) |
| $t_{IH}$ | Input hold time | 5 | — | ns | 25 pF ≥ $C_L$ (1 card) |
| $t_{O\_DLY\ (min)}$ | Output delay time during data transfer mode | 0 | 14 | ns | 25 pF ≥ $C_L$ (1 card) |
| $t_{O\_DLY\ (max)}$ | Output delay time during identification mode | 0 | 50 | ns | 25 pF ≥ $C_L$ (1 card) |

## 6.4 IO Description

Table 6-5 describes the SPI interface pins.

*Table 6-5.* **SPI Interface Pins**

| Pin Name | Pin | I/O | Description |
|---|---|---|---|
| SPI_CS | T1 | I | SPI chip select |
| SPI_CLK | V2 | I | Clock from SPI host |
| SPI_MOSI | R1 | I | Master out slave in |
| SPI_MISO | U1 | O | Master in slave out |
| SPI_INT | T2 | O | PI host interface output |

## 6.5 SPI Timing Flow

### 6.5.1 PIO Writes

Figure 6-5 displays 8-bit PIO writes (status check after data phase is optional).

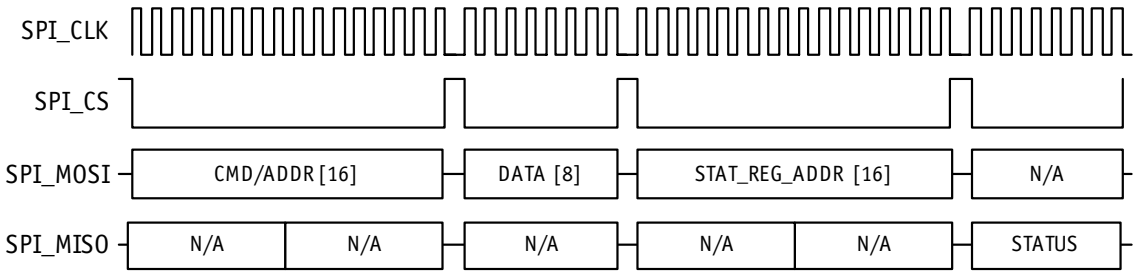



*Figure 6-5.* **PIO Write in 8-Bit Data Mode**

Figure 6-6 displays PIO writes in 16-bit data mode (status check phase is not shown).

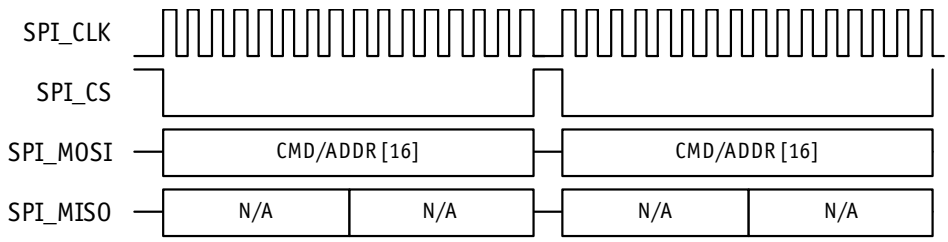



*Figure 6-6.* **PIO Write in 16-Bit Data Mode**

Figure 6-7 displays PIO writes in 32-bit data mode (status check phase is not shown).

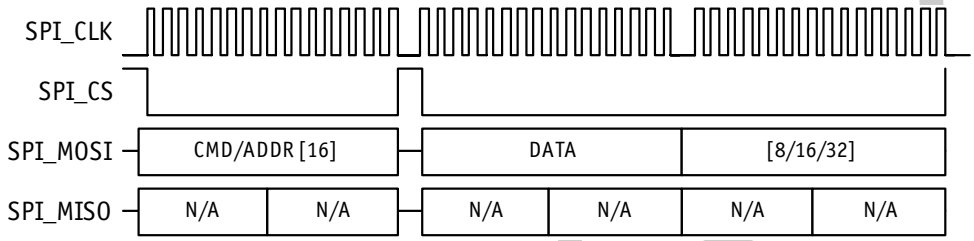



*Figure 6-7.* **PIO Write in 32-Bit Data Mode**

### *6.5.2 PIO Reads*

Figure 6-8 displays PIO reads in 8-bit data mode.

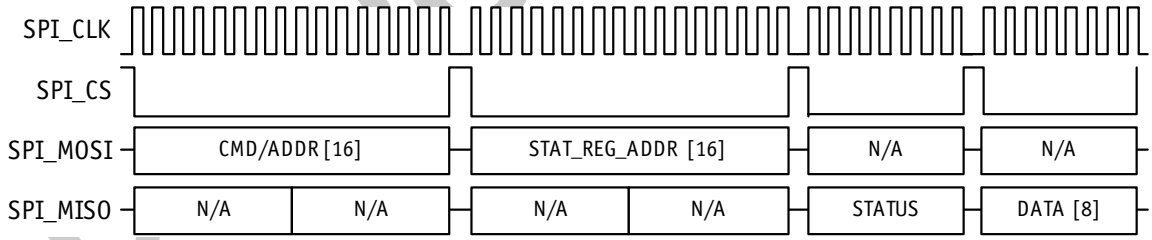



*Figure 6-8.* **PIO Read in 8-Bit Data Mode**

Figure 6-9 displays PIO reads in 16-bit data mode.

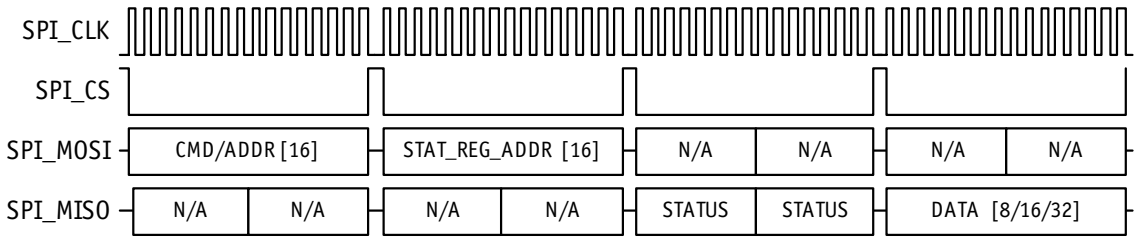



*Figure 6-9.* **PIO Read in 16-Bit Data Mode**

Figure 6-9 displays 32-bit data mode PIO reads.

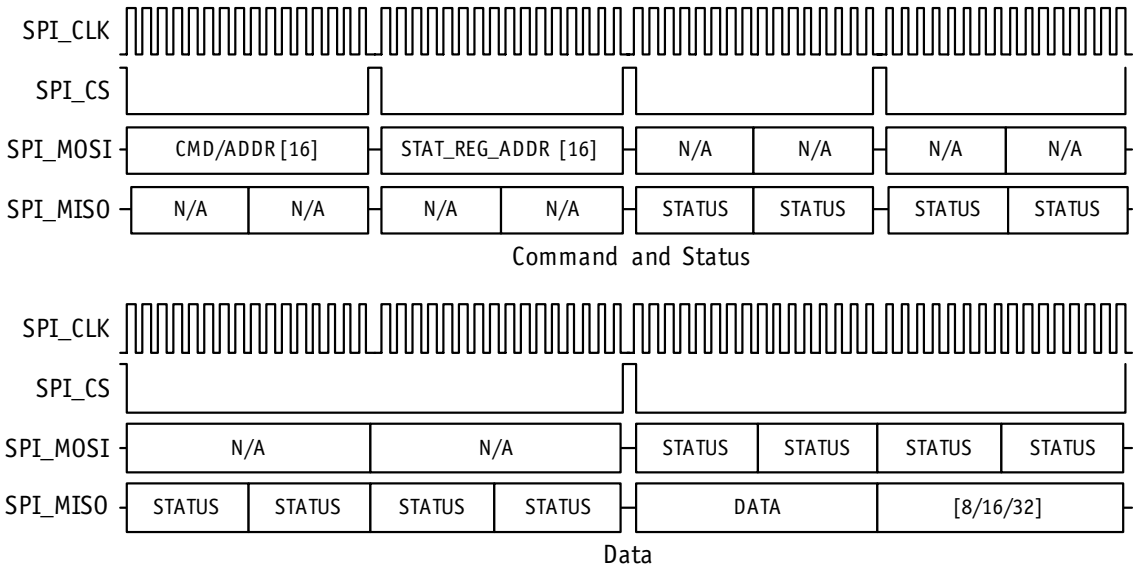



*Figure 6-10.* **PIO Read in 32-Bit Data Mode**

### *6.5.3 DMA Writes*

These steps are required for DMA burst/write.

1. Program the required configuration (Endianness, data size, etc.) by writing to the CONFIG register.
2. Program the DMA address into the SPI_DMA_ADDR register using the PIO write command.
3. Program the number of bytes to transfer into the DMA into SPI_DMA_COUNT and set the DMA_EN bit in the same register.
4. Start shifting the DATA. SPI_DATA_SIZE bits in the SPI_CONFIG register determine the width of each transfer.
5. On shifting the required number of bytes, poll the status register bit 0 using PIO reads from SPI_STATUS_REG, until the status register bit 0 is set to 1.

Figure 6-11 shows a DMA write 16-bit example for steps 2 and 3, programming the DMA address and count.

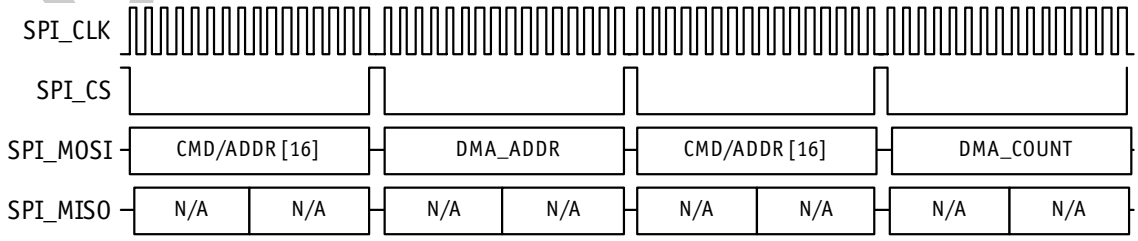



*Figure 6-11.* **Example: Programming the DMA Address and Count**

Figure 6-12 shows a DMA write 16-bit example for the write data phases.

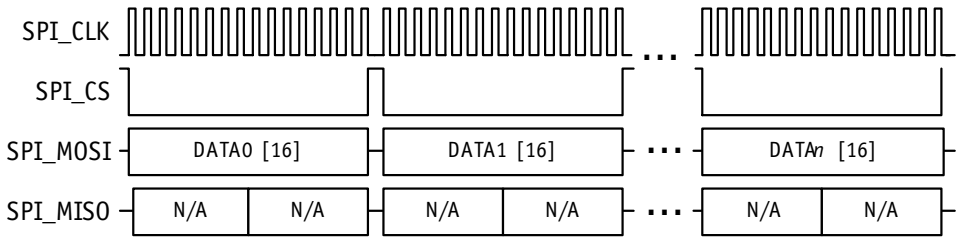



*Figure 6-12.* **Example: Write Data Phases**

Figure 6-13 shows a DMA write 16-bit example for check status phase.

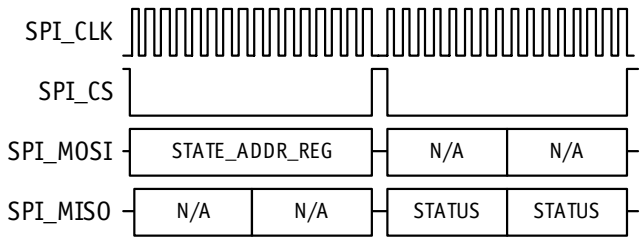



*Figure 6-13.* **Example: Check Status Phase**



### *6.5.4 DMA Reads*

1. Program the required configuration (Endianness, data size, etc.) by writing to the CONFIG register.
2. Program the DMA address into the SPI_DMA_ADDR register using the PIO write command.
3. Program the number of bytes to transfer to the DMA in SPI_DMA_COUNT and set the DMA_EN bit in the same register.
4. Poll the status register bit 0 using PIO reads of the SPI_STATUS_REG until the until the status register bit 0 is set to 1.
5. Start reading the DATA. SPI_DATA_SIZE bits in the SPI_CONFIG register determine the width of each transfer.

Figure 6-11 shows a DMA read 16-bit example for steps 2 and 3, programming the DMA address and count. Figure 6-14 shows a DMA write 16-bit example for the write data phases.

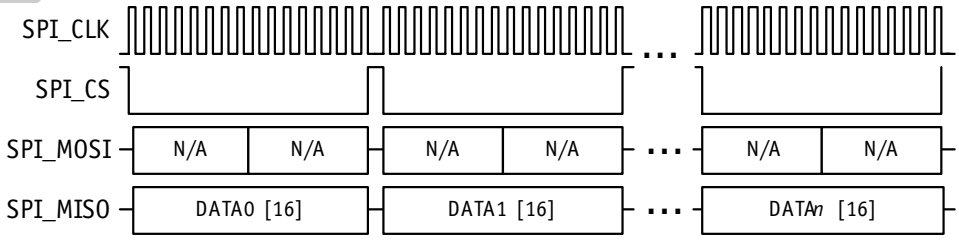



*Figure 6-14.* **Example: Read Data Phases**

Figure 6-13 shows a DMA write 16-bit example for check status phase. The DATA payload must be transferred to/from the SPI host. The width of each transfer depends on the value of the SPI_DATA_SIZE bits.

The supported data size is 8/16/32. When the host reads the SPI_STATUS register, the contents of the register shift out on the SPI_MISO output.

### 6.6 Error Recovery

If any errors occur, the SPI core stops further accesses to mailbox/internal registers and buffer RAMs. SPI core waits without any active operation in a safe mode until the last data phase of the current transaction, then moves into a state where only status register reads/ writes are accepted. Upon receiving the error INTR assertion, host is expected to write and clear the error condition by writing to the status register as given above. Only then further normal transactions can proceed.

### 6.7 Early Transaction Termination

These exceptions apply to the data size set in the data size register.

Read

- 8 CLK data phase in DATA16 or DATA32 mode will be treated as an 8-bit read
- 16 CLK data phase in DATA32 mode will be treated as a 16-bit read
- 24 CLK data phase in DATA32 mode will be treated as a 24-bit read

Write

- 8 CLK data phase in DATA16 or DATA32 mode will be treated as a 8-bit write
- 16 CLK data phase in DATA32 mode will be treated as a 16-bit write
- 24 CLK data phase in DATA32 mode will be treated as a 24-bit write

Early transaction termination is applicable for all mailbox register accesses and the last transfer of a DMA request in DATA16 and DATA32 modes. Internal registers are accessed only using 16 CLK DATA phase in DATA16 and DATA32 modes, and using 8 CLK DATA phase in DATA8 mode.

The local bus register (at 16'h0470) is always accessed using the 16 CLK data phase. Also note that due to the Endian organization of the bytes the early transaction termination in Big Endian mode might result in missing data bytes unless the correct byte/word address is given (e.g., to read the byte at address 14'h0101, issuing a DATA16 transaction at 14'h0101 and termination at Read8 will not work. The correct method is to use address 14'h0101 and do a read8 termination).

### 6.8 Interrupts

An interrupt is asserted on the SPI_INTR output port of the SPI slave on one of these conditions. The interrupt asserts only if the INTR_ENABLE bit is set in the SPI configuration register, or SPI_INTR appears on the SDDAT[2] pin of the chip and can be used to interrupt the host for these conditions:

- DMA Completion
- SPI Interface error
- Address error
- Read error
- Write error

The corresponding bit in the STATUS register is set to 1 to reflect the appropriate error or event. Upon reading the status register, the host is expected to clear the interrupt condition by writing back to the status register with the appropriate bit set to 1.

### 6.9 32-Bit Operation

The SPI Slave does not support 8-bit write terminations (WR8) during single writes in DATA32 mode. DMA writes in DATA32 mode, however, can do WR8 for the last transfer in a DMA. Host software can do one of the following if 8_bit single writes are required:

- In DATA32 mode when required to do WR8
  - Change data mode to DATA8
  - Perform the WR8 transaction
  - Revert back to DATA32 mode and continue further transactions
- In DATA32 mode when required to do WR8
  - Change data mode to DATA16
  - Perform WR8 or WR16 as required
  - Revert back to DATA32 mode and continue further transactions
- An optimal method recommended for OMAP host is to be in default as DATA16 mode. Most operations can be done efficiently in 16-bit mode. Many SPI slave internal registers are also in 16-bit mode. For large (DMA) transfers, SPI can be put in 32-bit mode, and the DMA can be done in DATA32 mode.

### *6.10 Clock Frequency Selection*

In general, the SPI_CLK used in the SPI interface can be independent (asynchronous) to the core clock (SYS_CLK) used in the rest of the chip. However, some restrictions exist on the frequency of the SPI_CLK clock input signal:

- The SPI_CLK frequency cannot be more than 1.5 times the Core Clock. For example, if the Core Clock (SYS_CLK) is 20 MHz, the SPI_CLK cannot be more than 30 MHz for proper operation.
- For operating SPI_CLK at low frequencies (< 5 MHz) a special mode bit has to be set. This enables certain internal logic for tighter synchronization between the low frequency SPI_CLK and the SYS _CLK. The bit to set is bit 0 of the special configuration register (at internal SPI register address 16'h4800).

# 7. Register Descriptions

This section describes internal registers for the various blocks of the AR6001X.

## 7.1 RTC Block Registers

Table 7-1 summarizes RTC block registers.

*Table 7-1.* **RTC Block Register Summary**

| Offset | Name | Description | Page |
|---|---|---|---|
| 0x0C000000 | RESET_CONTROL | Controls individual reset pulses to functional blocks | page 54 |
| 0x0C000004 | XTAL_CONTROL | Controls the analog crystal interface | page 54 |
| 0x0C000008 | TCXO_DETECT | Detects presence of an external TCXO device | page 56 |
| 0x0C000014 | PLL_CONTROL | Control settings for the PLL | page 56 |
| 0x0C000018 | PLL_SETTLE | Sets the PLL settling time | page 56 |
| 0x0C00001C | XTAL_SETTLE | Sets the crystal settling time | page 57 |
| 0x0C000020 | CORE_CLOCK | Controls the core clock speed | page 57 |
| 0x0C000024 | CPU_CLOCK | Controls the CPU clock speed | page 58 |
| 0x0C00002C | CLOCK_CONTROL | Controls clock gating to individual functional blocks | page 58 |
| 0x0C000034 | REF_VOLTAGE_TRIM | Controls the reference voltage trim | page 58 |
| 0x0C000038 | LDO_CONTROL | Controls the on chip LDOs | page 59 |
| 0x0C00003C | WDT_CONTROL | Watchdog timer actions | page 59 |
| 0x0C000040 | WDT_STATUS | Watchdog timer interrupt status | page 59 |
| 0x0C000044 | WDT | Watchdog timer compare target | page 60 |
| 0x0C000048 | WDT_COUNT | Watchdog timer current count | page 60 |
| 0x0C00004C | WDT_RESET | Watchdog timer reset | page 60 |
| 0x0C000050 | INT_STATUS | Interrupt Status | page 61 |
| 0x0C000054 | LF_TIMER0 | Low frequency timer 0 compare target | page 61 |
| 0x0C000058 | LF_TIMER_COUNT0 | Current Low frequency timer count | page 62 |
| 0x0C00005C | LF_TIMER_CONTROL0 | Low frequency timer 0 control bits | page 62 |
| 0x0C000060 | LF_TIMER_STATUS0 | Low frequency timer interrupt status | page 62 |
| 0x0C000064 | LF_TIMER1 | Low frequency timer 1 compare target | page 63 |
| 0x0C000068 | LF_TIMER_COUNT1 | Low frequency timer current count | page 63 |
| 0x0C00006C | LF_TIMER_CONTROL1 | Low frequency timer 1 control bits | page 63 |
| 0x0C000070 | LF_TIMER_STATUS1 | Low frequency timer interrupt status | page 64 |
| 0x0C000074 | LF_TIMER2 | Low frequency timer compare target | page 64 |
| 0x0C000078 | LF_TIMER_COUNT2 | Low frequency timer current count | page 64 |
| 0x0C00007C | LF_TIMER_CONTROL2 | Low frequency timer current count | page 64 |
| 0x0C000080 | LF_TIMER_STATUS2 | Low frequency timer interrupt status | page 65 |
| 0x0C000084 | LF_TIMER3 | Low frequency timer compare target | page 65 |
| 0x0C000088 | LF_TIMER_COUNT3 | Low frequency timer current count | page 65 |
| 0x0C00008C | LF_TIMER_CONTROL3 | Low frequency timer current count | page 65 |
| 0x0C000090 | LF_TIMER_STATUS3 | Low frequency timer interrupt status | page 66 |
| 0x0C000094 | HF_TIMER | High frequency timer compare target | page 66 |
| 0x0C000098 | HF_TIMER_COUNT | High frequency timer current count | page 67 |
| 0x0C00009C | HF_LF_COUNT | Captured low frequency timer value relative to the high frequency timer read | page 67 |
| 0x0C0000A0 | HF_TIMER_CONTROL | High frequency control bits | page 67 |
| 0x0C0000A4 | HF_TIMER_STATUS | High frequency timer interrupt status | page 68 |
| 0x0C0000A8 | RTC_CONTROL | Loads RTC values into RTC logic | page 68 |
| 0x0C0000AC | RTC_TIME | RTC time of day | page 68 |
| 0x0C0000B0 | RTC_DATE | RTC date and year | page 69 |
| 0x0C0000B4 | RTC_SET_TIME | RTC set time of day | page 69 |
| 0x0C0000B8 | RTC_SET_DATE | RTC set day and year | page 69 |
| 0x0C0000BC | RTC_SET_ALARM | RTC alarm time of day | page 70 |
| 0x0C0000C0 | RTC_CONFIG | RTC operation configuration | page 70 |
| 0x0C0000C4 | RTC_ALARM_STATUS | Enable, set, and clear RTC alarm interrupt | page 71 |
| 0x0C0000C8 | UART_WAKEUP | Enable UART wakeup events | page 71 |

*Table 7-1.* **RTC Block Register Summary (continued)**

| Offset | Name | Description | Page |
|---|---|---|---|
| 0x0C0000CC | RESET_CAUSE | Reset cause | page 71 |
| 0x0C0000D0 | SYSTEM_SLEEP | System sleep status bits | page 73 |
| 0x0C0000D4 | LDO_VOLTAGE | LDO_D voltage control | page 73 |
| 0x0C0000D8 | LDO_A_VOLTAGE | LDO_A voltage control | page 74 |
| 0x0C0000DC | SDIO_LDO_VOLTAGE | SDIO_LDO voltage control | page 74 |
| 0x0C0000E0 | CORE_PAD_ENABLE | Core pad enable control | page 75 |
| 0x0C0000E4 | SDIO_WRAPPER | SDIO signal wrapper control | page 75 |
| 0x0C0000E8 | MAC_SLEEP_CONTROL | MAC sleep options control | page 75 |
| 0x0C0000EC | KEEP_AWAKE | Keep awake timer | page 75 |
| 0x0C0000F0 | CHIP_REV | Chip Rev ID | page 76 |
| 0x0C0000F4 | DERIVED_RTC_CLK | 32 KHz HF clock creation | page 76 |
| 0x0C0000F8 | ACG_DISABLE | Automatic clock gating control | page 76 |

### *7.1.1 Reset Control (RESET_CONTROL)*

Offset: 0x0C000000
Reset Value: 0x0
Access: Read/Write

Software can hold any target block in reset by writing a 1 to the corresponding bit in this register. Reset is held asserted to the target block as long as the corresponding bit is set. Multiple blocks may be held in reset simultaneously.

| Bit | Bit Name | Description |
|---|---|---|
| 31:10 | RES | Reserved |
| 9 | RST_OUT | Asserts the RST_OUT_L pin. Note that RST_OUT_L also asserts during SYS_RST_L and COLD_RESET.<br>■ 1 = Drive the RST_OUT_L pin to 0<br>■ 0 = Drive the RST_OUT_L pin to 1 |
| 8 | COLD_RST | Resets all AR6001X blocks with Cold Reset. This process completes in around 1 second, and the bit clears automatically when chip reset completes. |
| 7 | WARM_RST | Sends a WARM_RESET to all AR6001X blocks that support WARM_RESET (MAC and CPU blocks only). This bit clears automatically after the warm reset process completes. |
| 6 | CPU_WARM_RST | Resets the CPU block only. Reset type is Warm Reset. Support blocks, including the memory controller and interrupt controller blocks, are not reset. This bit clears automatically after the CPU reset process completes. |
| 5 | MAC_COLD_RST | Holds MAC block in cold reset, including the baseband and radio. |
| 4 | MAC_WARM_RST | Holds MAC block in warm reset, including the baseband and radio. |
| 3 | RES | Reserved |
| 2 | MBOX_RST | Holds MBOX block in reset. |
| 1 | UART_RST | Holds UART block in reset. |
| 0 | SI0_RST | Holds serial interface (SPI and $I^2C$) logic block in reset. |

### *7.1.2 Crystal Control (XTAL_CONTROL)*

Offset: 0x0C000004
Reset Value: 0x0
Access: Read/Write

This register controls the regulator and the clock source selection between an TCXO and a crystal.

| Bit | Bit Name | Description |
|---|---|---|
| 31:1 | RES | Reserved |
| 0 | TCXO | ■ 1 = The chip is being driven by a TCXO device<br>■ 0 = The chip is being driven by a crystal<br>Note that when a TCXO device is used, software should set this field to 1<br>WARNING: If this field is set to 1 when a crystal is being used, the high speed clock stops and the chip hangs. |

### 7.1.3 TCXO Detection (TCXO_DETECT)

Offset: 0x0C000008
Reset Value: 0x0
Access: Read only

This register returns the value of the TCXO detection circuitry. This value is only meaningful when XTAL_CONTROL_TCXO=0. If software detects that a TCXO is being used, it should set XTAL_CONTROL_TXCO to 1.

| Bit | Bit Name | Description |
|---|---|---|
| 31:1 | RES | Reserved |
| 0 | PRESENT | ■ 1 = A TCXO device is detected<br>■ 0 = no TCXO detected |

### 7.1.4 PLL Control (PLL_CONTROL)

Offset: 0x0C000014
Reset Value: 0x0
Access: Read/Write

This register provides access to PLL setup control signals. Any writes to this register freeze all high speed clocks for 61 µs: the clock select lines and PLL control lines change after 30.5 µs, then another 30.5 µs passes before allowing clocks to settle.

PLL freq = (reFClk/refdiv) * div[8:0] / (2*(div[9] + 1))

| Bit | Bit Name | Description |
|---|---|---|
| 31:21 | RES | Reserved |
| 20 | DIG_TEST_CLK | Bypasses PLL, uses DIG_TEST_CLK input (test mode only) |
| 19 | MAC_OVERRIDE | When set, a MAC clock request deasserts pllbypass even if the BYPASS field is set to 1. The bit can be set when it is desirable for the SOC_ON state to not use the PLL, but the ON state to use the PLL. |
| 18 | NOPWD | Prevents the PLL from being powered down when pllbypass is asserted. Set only for testing purposes. |
| 17 | UPDATING | Set during the PLL update process. After software writes PLL_CONTROL, it takes about 45 µs for the update to occur. Software may poll the bit to see if the update has taken place.<br>■ 1 = PLL update is pending<br>■ 0 = PLL update is complete |
| 16 | BYPASS | Bypass PLL. Defaults to 1 for test purposes; software must enable the PLL for normal operation. |
| 15:12 | REFDIV | Reference clock divider. |
| 11:10 | RES | Reserved |
| 9:0 | DIV | Primary multiplier. MSB is divide by 2 factor. |

### 7.1.5 PLL Settle Time (PLL_SETTLE)

Offset: 0x0C000018
Reset Value: 0x400
Access: Read/Write

The PLL requires time to settle once powered up or reprogrammed. Each time the PLL parameters change due to a write to the PLL register or to a system event that changes PLL control, hardware gates off the clocks for PLL_SETTLE time while the PLL stabilizes. Units are in reFClk periods.

| Bit | Bit Name | Description |
|---|---|---|
| 31:11 | RES | Reserved |
| 10:0 | TIME | Time required for the PLL to settle. Units are in reFClk periods, so the default value of 1024 results in a 25.6 µs settling time. This register should never be set under 100. |

### 7.1.6 Crystal Settle Time (XTAL_SETTLE)

Offset: 0x0C00001C
Reset Value: 0x42
Access: Read/Write

The external crystal requires time to settle once powered up, which occurs as the chip passes through the WAKEUP state, between OFF and ON or between SLEEP and ON. The exact time varies, so the register allows XTAL power up FSM transitions in the minimal correct time. The default value of 63 always allows the XTAL to fully settle before clocks are enabled, but it can be set to a smaller value if hardware characterization approves. The timer expires in (XTAL_SETTLE + 1) clocks. XTAL_SETTLE retains its programmed value in the RTC block during reset.

The value programmed in this register should match the MAC register Sleep Clock 32 KHz Wake field SLEEP32_WAKE_XTL_TIME. Note that the MAC register value is in ms.

| Bit | Bit Name | Description |
|---|---|---|
| 31:7 | RES | Reserved |
| 6:0 | TIME | Time required for the XTAL to settle. Units are in 30 µs, so the default value of 66 results in 2.0 ms settling time. This register should never be set to 0. |

### 7.1.7 Core Clock (CORE_CLOCK)

Offset: 0x0C000020
Reset Value: 0x42
Access: Read/Write

When this register is written to, the core clock is gated for two high speed clock cycles while the clock dividers are updated. The core clock drives the AHB and memory controller.

| Bit | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12 | DIG_TEST | Causes the digital clock tree to be driven from the DIG_TEST_CLK input pin. This mode is for test purposes only, and should never be set during normal operation.<br>■ 0 = Normal operation, use XTAL_OUT pin for digital clocks<br>■ 1 = Test operation, use DIG_TEST_CLK input for digital clocks |
| 11:10 | RES | Reserved |
| 9:8 | STANDARD | Controls the CORE speed during standard operation. The CORE clock speed updates on the clock following the write to this register.<br>■ 0 = 20/22 MHz<br>■ 1 = 40/44 MHz<br>■ 2 = 80/88 MHz<br>■ 3 = reFClk |
| 7:2 | RES | Reserved |
| 1:0 | REDUCED | Controls the CORE speed during reduced power operation. Reduced power operation occurs when the CORE sets the RP bit in the CP0 Status, and the EXL and ERL bits are cleared. In debug mode, the standard speed is used. The CORE clock speed updates on the clock following the write to this register.<br>■ 0 = 20/22 MHz<br>■ 1 = 40/44 MHz<br>■ 2 = 80/88 MHz<br>■ 3 = refClk |

### 7.1.8 CPU Clock (CPU_CLOCK)

Offset: 0x0C000024
Reset Value: 0x00
Access: Read/Write

When this register is written to, the CPU clock is gated for two high speed clock cycles while the clock dividers are updated.

| Bit | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12 | DISABLE_SYNC | When this field is cleared, the CPU synchronization FIFOs will be bypassed if the CPU and CORE clocks are equal. This reduced CPU access latencies.<br>■ 0 = Sync FIFO bypass enabled when CPU clock and core clock are equal<br>■ 1 = Sync FIFO bypass disabled, always use async |
| 11:10 | RES | Reserved |
| 9:8 | STANDARD | Controls the CPU speed during standard operation. The CPU clock speed updates on the clock following the write to this register.<br>■ 0 = 40/44 MHz<br>■ 1 = 80/88 MHz<br>■ 2 = 107/117 MHz<br>■ 3 = reFClk |
| 7:2 | RES | Reserved |
| 1:0 | REDUCED | Controls the CPU speed during reduced power operation. Reduced power operation occurs when the CPU sets the RP bit in the CP0 Status, and the EXL and ERL bits are cleared. In debug mode, the standard speed is used. The CPU clock speed updates on the clock following the write to this register.<br>■ 0 = 40/44 MHz<br>■ 1 = 80/88 MHz<br>■ 2 = 107/117 MHz<br>■ 3 = reFClk |

### 7.1.9 Clock Gating Control (CLOCK_CONTROL)

Offset: 0x0C00002C
Reset Value: 0x1
Access: Read/Write

Software can gate off the clock to an individual functional block to save power. Note that when a functional block has it clock gated, it cannot wake up, cause interrupts, or operate until its clock gate has been released.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | UART_CLK | Set to 1 to gate off clock to UART block |
| 0 | SI0_CLK | Set to 1 to gate off clock to SI0 logic block |

### 7.1.10 Reference Voltage Trim Control (REF_VOLTAGE_TRIM)

Offset: 0x0C000034
Reset Value: 0x8
Access: Read/Write

This register holds the trim setting for the on-chip 1.8V reference which is calibrated at manufacturing time. The value should be programmed at system boot time with the calibration value stored in flash.

| Bits | Bit Name | Description |
|---|---|---|
| 31:4 | RES | Reserved |
| 3:0 | REFSEL | Trim bits for 1.8V reference voltage |

### 7.1.11 On-Chip LDO Control (LDO_CONTROL)

Offset: 0x0C000038
Reset Value: 0x0
Access: Read/Write

The on chip LDO control inputs can be controlled by writing to this register. These fields return to their default values after every reset event.

| Bits | Bit Name | Description |
|---|---|---|
| 31:15 | RES | Reserved |
| 14 | CORE_LIMIT_OFF | Sets current limit mode for core LDO.<br>■ 0 = Current limit is on.<br>■ 1 = Current limit is off. |
| 13:11 | CORE_LIMIT | Controls short-circuit current limit. Default value is 4 = 400 mA. |
| 10:8 | CORE_REG_Z | Controls LDO stability. Default value is 3 (65° PM with 2.2 µF load cap at 10 mA). |
| 7 | RES | Reserved |
| 6 | RADIO_LIMIT_OFF | Sets current limit mode for radio LDO.<br>■ 0 = Current limit is on.<br>■ 1 = Current limit is off. |
| 5:3 | RADIO_LIMIT | Controls short-circuit current limit. Default value is 4 = 400 mA. |
| 2:0 | RADIO_REG_Z | Controls LDO stability. Default value is 3 (65 degree PM with 2.2 µF load cap at 10 mA). |

### 7.1.12 Watchdog Timer (WDT_CONTROL)

Offset: 0x0C00003C
Reset Value: 0x2
Access: Read/Write

Controls the watchdog timer actions.

| Bits | Bit Name | Description |
|---|---|---|
| 31:3 | RES | Reserved |
| 2:0 | ACTION | Control watchdog timer action on an expiration event:<br>■ 0xx = Watchdog actions disabled.<br>■ 100 = Watchdog reset action enabled, warm reset on expiration.<br>■ 101 = Watchdog NMI action enabled, NMI to CPU on expiration.<br>■ 110 = Watchdog interrupt action enabled on expiration.<br>■ 111 = reserved |

### 7.1.13 Watchdog Timer Interrupt Status (WDT_STATUS)

Offset: 0x0C000040
Reset Value: 0x0
Access: Read/Write

This signal asserts when WDT action is set to interrupt and a WDT expire event occurs.

| Bits | Bit Name | Description |
|---|---|---|
| 31:1 | RES | Reserved |
| 0 | INTERRUPT | Hardware sets this bit when WDT expires. Software can also set this bit with a write. When set by either hardware or software, the corresponding interrupt bit in INT_STATUS is set. |

### *7.1.14 Watchdog Timer Compare Target (WDT)*

Offset: 0x0C000044
Reset Value: 0x3FFFFF
Access: Read/Write

When the WDT_COUNT register equals the WDT register. The WDT logic takes the action specified by the WDT_CONTROL register. The WDT operates at 32 KHz.

| Bits | Bit Name | Description |
|---|---|---|
| 31:22 | RES | Reserved |
| 21:0 | TARGET | Watchdog timer target compare value, based on the core clock frequency. Software should reset the watchdog timer after changing this value. |

### *7.1.15 Watchdog Timer Current Count (WDT_COUNT)*

Offset: 0x0C000048
Reset Value: 0x0
Access: Read only

The current value of the watchdog timer. This value resets to 0 when the watchdog timer is reset. If the WDT_COUNT equals WDT, a watchdog expiration event occurs. The WDT_COUNT timer operates at the core clock frequency. When the core clock is gated off, the watchdog timer freezes.

| Bits | Bit Name | Description |
|---|---|---|
| 31:22 | RES | Reserved |
| 21:0 | VALUE | Watchdog timer current count value. Units are in 32.768 KHz clocks. |

### *7.1.16 Watchdog Timer Reset (WDT_RESET)*

Offset: 0x0C00004C
Reset Value: 0x0
Access: Read/Write

Resets the watchdog timer.

| Bits | Bit Name | Description |
|---|---|---|
| 31:1 | RES | Reserved |
| 0 | VALUE | This field is written by software periodically to reset the watchdog timer and prevent expiration under normal operation. When software writes a 1 to this field, the WDT_COUNT register is reset to 0. The watchdog timer does not expire until WDT_COUNT equals WDT.<br>When this field is read by software, a 0 is always returned. |

### 7.1.17 AR6001X CPU Interrupt Status (INT_STATUS)

Offset: 0x0C000050
Reset Value: 0x0
Access: Read only

The interrupt status for local CPU interrupts maps to MIPS hardware interrupts:

- IP7 = Interrupt timer in MIPS core
- IP6 = MAC interrupt
- IP5 = Mailbox interrupt
- IP4 = RTC timer interrupt (RTC alarm, HF_TIMER, or LF_TIMER*)
- IP3 = Peripheral Interrupt (SI, GPIO, UART, or keypad)
- IP2 = MC error interrupt or watchdog timer interrupt
- IP1 = Software interrupt (MIPS internal)
- IP0 = Software interrupt (MIPS internal)

Interrupts can be masked at this level using the MIPS CP0 Status register. Interrupt must be cleared at their source, and can usually be individually masked at the source as well.

| Bits | Bit Name | Description |
|---|---|---|
| 31:15 | RES | Reserved |
| 14 | TIMER | MIPS timer interrupt |
| 13 | MAC | Wireless MAC interrupt |
| 12 | MAILBOX | Host interface/mailbox interrupt |
| 11 | RTC_ALARM | RTC Alarm interrupt |
| 10 | HF_TIMER | High frequency timer interrupt |
| 9 | LF_TIMER3 | Low frequency timer 3 interrupt |
| 8 | LF_TIMER2 | Low frequency timer 2 interrupt |
| 7 | LF_TIMER1 | Low frequency timer 1 interrupt |
| 6 | LF_TIMER0 | Low frequency timer 0 interrupt |
| 5 | KEYPAD | Keypad interrupt, see KEY_STATUS for details. |
| 4 | SI | Serial Interface, I2S or SPI (master) interrupt |
| 3 | GPIO | GPIO Interrupt |
| 2 | UART | UART Interrupt |
| 1 | ERROR | Memory controller detected AHB or APB error. See memory controller registers for details. |
| 0 | WDT_INT | Watchdog Timeout Interrupt. See watchdog timer registers for details. |

### 7.1.18 LF Timer 0 Compare Target (LF_TIMER0)

Offset: 0x0C000054
Reset Value: 0x0
Access: Read/Write

When the LF_TIMER_COUNT0 register is equal to this register, a timer interrupt generates if enabled. Software can write to this register at any time, hardware performs the value synchronization across clock boundaries automatically after a write.

When the chip is in the SLEEP state, it wakes up before the LF_TIMER reaches its target. Hardware automatically causes a wakeup in XTAL_SETTLE cycles before the LF_TIMER expires. If XTAL_SETTLE is larger than the time remaining before LF_TIMER expires, the chip does not enter SLEEP.

If the SYSTEM_SLEEP_LIGHT register is set, early wakeup is not required, so the wakeup occurs when LF_TIMER_COUNT equals LF_TIMER_TARGET.

| Bits | Bit Name | Description |
|---|---|---|
| 31:0 | TARGET | Low frequency timer target compare value. Units are in 30.5 µs (1/32768 sec.). |

### *7.1.19 LF Timer 0 Current Count (LF_TIMER_COUNT0)*

Offset: 0x0C000058
Reset Value: 0x0
Access: Read only

The current low frequency timer value. This value is continuously synchronized from the timer clock domain to the core cock domain.

| Bits | Bit Name | Description |
|---|---|---|
| 31:0 | VALUE | Current low frequency timer count value. Units are in 30.5 µs (1/32768 sec.). |

### *7.1.20 LF Timer 0 Control Bits (LF_TIMER_CONTROL0)*

Offset: 0x0C00005C
Reset Value: 0x0
Access: Read/Write

Controls the low frequency timer 0 clock restart policy and reset.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | AUTO_RESTART | Timer automatic restart control.<br>■ 0 = LF Timer continues counting after LF_TIMER_COUNT0 reaches LF_TIMER0<br>■ 1 = LF Timer resets to 0 and continues counting after LF_TIMER_COUNT0 reaches LF_TIMER0. Because the timer resets to 0, the period of the timer is LF_TIMER+1 clocks. |
| 0 | RESET | Software writes a 1 to this field to reset the timer to 0. When software writes a 1, the timer begins counting from 0. Software can see this transition from 1 to 0 when the reset occurs. |

### *7.1.21 LF Timer 0 Interrupt Status (LF_TIMER_STATUS0)*

Offset: 0x0C000060
Reset Value: 0x0
Access: Read/Write

Low Frequency timer 0 raw interrupt and enable bits. This register holds the low frequency timer 0 interrupt status before any interrupt enable masks are applied.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | ENABLE | Enables the interrupt to propagate up to the INT_STATUS register and CPU. If the ENABLE bit is set to 0, this interrupt does not cause IP4 to be set. |
| 0 | INTERRUPT | Hardware sets this bit when 0 ≥ (LF_TIMER0 – LF_TIMER_COUNT0). Software can also set this bit with a write. When set by either hardware or software, the LF timer source interrupt signal asserts to the interrupt controller. Software clears the interrupt by writing 0 to this field. |

### *7.1.22 LF Timer 1 Compare Target (LF_TIMER1)*

Offset: 0x0C000064
Reset Value: 0x0
Access: Read/Write

When the LF_TIMER_COUNT1 register equals this register, a timer interrupt generates if enabled. Software can write to this register at any time, hardware performs the value synchronization across clock boundaries automatically after a write.

When the chip is in the SLEEP state, it wakes up before the LF_TIMER reaches its target. Hardware automatically causes a wakeup in XTAL_SETTLE cycles before the LF_TIMER expires. If XTAL_SETTLE is larger than the time remaining before LF_TIMER expires, the chip does not enter SLEEP.

If the SYSTEM_SLEEP_LIGHT register is set, early wakeup is not required, so the wakeup occurs when the LF_TIMER_COUNT equals LF_TIMER_TARGET

| Bits | Bit Name | Description |
|---|---|---|
| 31:0 | TARGET | Low frequency timer target compare value. Units are in 30.5 µs (1/32768 sec.). |

### *7.1.23 LF Timer 1 Current Count (LF_TIMER_COUNT1)*

Offset: 0x0C000068
Reset Value: 0x0
Access: Read only

The current value of the low frequency timer. This value is continuously synchronized from the timer clock domain to the core cock domain.

| Bits | Bit Name | Description |
|---|---|---|
| 31:0 | VALUE | Current low frequency timer count value. Units are in 30.5 µs (1/32768 sec.). |

### *7.1.24 LF Timer 1 Control Bits (LF_TIMER_CONTROL1)*

Offset: 0x0C00006C
Reset Value: 0x0
Access: Read only

Controls the clock restart policy and reset for low frequency timer 1.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | AUTO_RESTART | Timer automatic restart control.<br>■ 0 = LF Timer continues counting after LF_TIMER_COUNT0 reaches LF_TIMER0<br>■ 1 = LF Timer resets to 0 and continues counting after LF_TIMER_COUNT0 reaches LF_TIMER0. Because the timer resets to 0, the period of the timer is LF_TIMER+1 clocks. |
| 0 | RESET | Software writes a 1 to this field to reset the timer to 0. When software writes a 1, the timer begins counting from 0. Software can see this transition from 1 to 0 when the reset occurs. |

### *7.1.25 LF Timer 1 Interrupt Status (LF_TIMER_STATUS1)*

Offset: 0x0C000070
Reset Value: 0x0
Access: Read/Write

Low Frequency timer 1 raw interrupt and enable bits. This register holds the LF Timer 1 interrupt status before any interrupt enable masks are applied.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | ENABLE | Enables the interrupt to propagate up to the INT_STATUS register and CPU. If the ENABLE bit is set to 0, this interrupt does not cause IP4 to be set. |
| 0 | INTERRUPT | Hardware sets this bit when 0 ≥ (LF_TIMER1 – LF_TIMER_COUNT1). Software can also set this bit with a write. When set by either hardware or software, the LF timer source interrupt signal asserts to the interrupt controller. Software clears the interrupt by writing 0 to this field. |

### *7.1.26 LF Timer 2 Compare Target (LF_TIMER2)*

Offset: 0x0C000074
Reset Value: 0x0
Access: Read/Write

When the LF_TIMER_COUNT2 register equals this register, a timer interrupt generates if enabled. Software can write to this register at any time, hardware performs the value synchronization across clock boundaries automatically after a write.

When the chip is in the SLEEP state, it wakes up before the LF_TIMER reaches its target. Hardware automatically causes a wakeup XTAL_SETTLE cycles before the LF_TIMER expires. If XTAL_SETTLE is larger than the time remaining before LF_TIMER expires, the chip does not enter SLEEP.

If the SYSTEM_SLEEP_LIGHT register is set, early wakeup is not required, so wakeup occurs when LF_TIMER_COUNT equals LF_TIMER_TARGET.

| Bits | Bit Name | Description |
|---|---|---|
| 31:0 | TARGET | Low frequency timer target compare value. Units are in 30.5 µs (1/32768 sec.). |

### *7.1.27 LF Timer 2 Current Count (LF_TIMER_COUNT2)*

Offset: 0x0C000078
Reset Value: 0x0
Access: Read only

The current value of the low frequency timer. This value continuously synchronizes from the timer clock domain to the core cock domain.

| Bits | Bit Name | Description |
|---|---|---|
| 31:0 | VALUE | Current low frequency timer count value. Units are in 30.5 µs (1/32768 sec.). |

### *7.1.28 LF Timer 2 Control Bits (LF_TIMER_CONTROL2)*

Offset: 0x0C00007C
Reset Value: 0x0
Access: Read/Write

Controls the clock restart policy and reset for low frequency timer 2.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | AUTO_RESTART | Timer automatic restart control.<br>■ 0 = LF Timer continues counting after LF_TIMER_COUNT0 reaches LF_TIMER0<br>■ 1 = LF Timer resets to 0 and continues counting after LF_TIMER_COUNT0 reaches LF_TIMER0. Because the timer resets to 0, the period of the timer is LF_TIMER+1 clocks. |
| 0 | RESET | Software writes a 1 to this field to reset the timer to 0. When software writes a 1, the timer begins counting from 0. Software can see this transition from 1 to 0 when the reset occurs. |

### *7.1.29 LF Timer 2 Interrupt Status (LF_TIMER_STATUS2)*

Offset: 0x0C000080
Reset Value: 0x0
Access: Read/Write

Low Frequency timer 2 raw interrupt and enable bits. This register holds the LF Timer 2 interrupt status before any interrupt enable masks are applied.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | ENABLE | Enables the interrupt to propagate up to the INT_STATUS register and CPU. If the ENABLE bit is set to 0, this interrupt does not cause IP4 to be set. |
| 0 | INTERRUPT | Hardware sets this bit when 0 ≥ (LF_TIMER2 – LF_TIMER_COUNT2). Software can also set this bit with a write. When set by either hardware or software, the LF timer source interrupt signal asserts to the interrupt controller. Software clears the interrupt by writing 0 to this field. |

### *7.1.30 LF Timer 3 Compare Target (LF_TIMER3)*

Offset: 0x0C000084
Reset Value: 0x0
Access: Read/Write

When the LF_TIMER_COUNT3 register equals this register, a timer interrupt is generated if enabled. Software can write to this register at any time, hardware will perform the value synchronization across clock boundaries automatically after a write.

When the chip is in the SLEEP state, it wakes up before the LF_TIMER reaches its target. Hardware automatically causes a wakeup XTAL_SETTLE cycles before the LF_TIMER expires. If XTAL_SETTLE is larger than the time remaining before LF_TIMER expires, the chip will not enter SLEEP.

If the SYSTEM_SLEEP_LIGHT register is set, early wakeup is not required, so the wakeup will occur when LF_TIMER_COUNT equals LF_TIMER_TARGET.

| Bits | Bit Name | Description |
|---|---|---|
| 31:0 | TARGET | Low frequency timer target compare value. Units are in 30.5 µs (1/32768 sec.). |

### *7.1.31 LF Timer 3 Current Count (LF_TIMER_COUNT3)*

Offset: 0x0C000088
Reset Value: 0x0
Access: Read only

The current value of the low frequency timer. This value is continuously synchronized from the timer clock domain to the core cock domain.

| Bits | Bit Name | Description |
|---|---|---|
| 31:0 | VALUE | Current low frequency timer count value. Units are in 30.5 µs (1/32768 sec.). |

### *7.1.32 LF Timer 3 Control Bits (LF_TIMER_CONTROL3)*

Offset: 0x0C00008C
Reset Value: 0x0
Access: Read/Write

Controls the clock restart policy and reset for low frequency timer 3.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | AUTO_RESTART | Timer automatic restart control.<br>■ 0 = LF Timer continues counting after LF_TIMER_COUNT0 reaches LF_TIMER0<br>■ 1 = LF Timer resets to 0 and continues counting after LF_TIMER_COUNT0 reaches LF_TIMER0. Because the timer resets to 0, the period of the timer is LF_TIMER+1 clocks. |
| 0 | RESET | Software writes a 1 to this field to reset the timer to 0. When software writes a 1, the timer begins counting from 0. Software can see this transition from 1 to 0 when the reset occurs. |

### *7.1.33 LF Timer 3 Interrupt Status (LF_TIMER_STATUS3)*

Offset: 0x0C000090
Reset Value: 0x0
Access: Read/Write

Low frequency timer 3 raw interrupt and enable bits. This register holds the low frequency timer 3 interrupt status before any interrupt enable masks are applied.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | ENABLE | Enables the interrupt to propagate up to the INT_STATUS register and CPU. If the ENABLE bit is set to 0, this interrupt does not cause IP4 to be set. |
| 0 | INTERRUPT | Hardware sets this bit when 0 ≥ (LF_TIMER3 – LF_TIMER_COUNT3). Software can also set this bit with a write. When set by either hardware or software, the LF timer source interrupt signal asserts to the interrupt controller. Software clears the interrupt by writing 0 to this field. |

### *7.1.34 HF Timer Compare Target (HF_TIMER)*

Offset: 0x0C000094
Reset Value: 0x0
Access: Read/Write

When the HF_TIMER_COUNT register equals this register, a timer interrupt generates if enabled. Software can write to this register at any time. The HF_TIMER does not run when the chip is in sleep mode and the high frequency clock is gated off.

| Bits | Bit Name | Description |
|---|---|---|
| 31:12 | TARGET | High frequency timer target compare value. Units are in 40 MHz clocks.<br>Note: The value is left justified so software can use the sign bit for wrap detection. |
| 11:0 | RES | Reserved |

### 7.1.35 HF Timer current count. (HF_TIMER_COUNT)

The current value of the high frequency timer.

Offset: 0x0C000098
Reset Value: 0x0
Access: Read only

| Bits | Bit Name | Description |
|---|---|---|
| 31:12 | VALUE | High frequency timer current count value. Units are in 40 MHz clocks. |
| 11:0 | REF | Reserved |

### 7.1.36 Captured LF Timer Value Relative to HF Timer Read (HF_LF_COUNT)

When software reads the HF_TIMER_COUNT register, hardware automatically copies the LF_TIMER0_COUNT value to this register, allowing software to capture both high and low frequency counter at the same instant in time

Offset: 0x0C00009C
Reset Value: 0x0
Access: Read only

| Bits | Bit Name | Description |
|---|---|---|
| 31:0 | VALUE | Low frequency timer, captured on last read to HF_TIMER_COUNT. Units are in 30.5 µs (1/32768 sec.). |

### 7.1.37 .HF Timer Control Bits (HF_TIMER_CONTROL)

Controls the timer enable, clock restart policy, and reset for HF_TIMER.

Offset: 0x0C0000A0
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:3 | RES | Reserved |
| 2 | ENABLE | Enables the timer operation. When not in use, timer should be disabled to save power.<br>■ 0 = High frequency timer is disabled<br>■ 1 = High frequency timer is enabled |
| 1 | AUTO_RESTART | Timer automatic restart control.<br>■ 0 = High frequency timer continues counting after HF_TIMER_COUNT reaches HF_TIMER<br>■ 1 = High frequency timer resets to 0 and continue counting after HF_TIMER_COUNT reaches HF_TIMER. Because the timer resets to 0, the period of the timer is HF_TIMER+1 clocks |
| 0 | RESET | Software writes a 1 to this field to reset the timer to 0. When software writes a 1, the timer begins counting from 0. |

### *7.1.38 HF Timer Interrupt Status (HF_TIMER_STATUS)*

Offset: 0x0C0000A4
Reset Value: 0x0
Access: Read/Write

High Frequency timer raw interrupt bit. This register holds the high frequency timer interrupt status before any interrupt enable masks are applied.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | ENABLE | This bit enables the interrupt to propagate up to the INT_STATUS register and to the CPU. If the ENABLE bit is set to 0, this interrupt does not cause IP4 to be set. |
| 0 | INTERRUPT | When HF_TIMER = HF_TIMER_COUNT, hardware sets this bit. Software can also set this bit with a write. When set by either hardware or software, the HF timer source interrupt signal is asserted to the interrupt controller. Software clears the interrupt by writing 0 to this field. |

### *7.1.39 RTC Values Load into RTC Logic (RTC_CONTROL)*

Offset: 0x0C0000A8
Reset Value: 0x0
Access: Read/Write

Controls loading of RTC config into the RTC logic. This register is only reset by RTC_RESET.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | LOAD_RTC | When software writes a 1 to this field, The RTC logic is loaded with the RTC_SET_TIME, RTC_SET_DATE, and RTC_CONFIG registers. For accurate clock setting, the sub-second count will be set to 0 when the write occurs.<br>When this field is read by software, a 1 is returned while the load is in progress, a 0 is returned when the load is complete. |
| 0 | LOAD_ALARM | When software writes a 1 to this field, The RTC alarm logic is loaded with the RTC_SET_ALARM register.<br>When this field is read by software, a 1 is returned while the load is in progress, a 0 is returned when the load is complete. |

### *7.1.40 RTC Time of Day (RTC_TIME)*

Offset: 0x0C0000AC
Reset Value: See field descriptions
Access: Read only

Returns current time values. This register is continuously synchronized by hardware to the core clock domain.

This register is only reset by RTC_RESET.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31:27 | RES | 0x0 | Reserved |
| 26:24 | WEEK_DAY | 0x1 | Set the current weekday, which changes at midnight; number rolls after 7 (1=Sunday, 2=Monday, etc.). |
| 23:22 | RES | 0x0 | Reserved |
| 21:16 | HOUR | 0x12 | Set the current hour 0–23 in 24-hour mode or 0–12 AM/PM in twelve-hour mode; resets to midnight, BCD. |
| 15 | RES | 0x0 | Reserved |
| 14:8 | MINUTE | 0x0 | Set the current minute count; rolls after 59 |
| 7 | RES | 0x0 | Reserved |
| 6:0 | SECOND | 0x0 | Set the current second count; rolls after 59 |

### 7.1.41 RTC Date and Year (RTC_DATE)

Offset: 0x0C0000B0
Reset Value: See field descriptions
Access: Read only

Returns current time values. This register is continuously synchronized by hardware to the core clock domain.

This register is only reset by RTC_RESET.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31:24 | RES | 0x0 | Reserved |
| 23:16 | YEAR | 0x0 | Set the current year (0 = 20000) |
| 15:13 | RES | 0x0 | Reserved |
| 12:8 | MONTH | 0x1 | Set the current month count; rolls after 12 (1=January, 2=February, etc.) |
| 7:6 | RES | 0x0 | Reserved |
| 5:0 | MONTH_DAY | 0x1 | Set the current day of the month, rolls at 28 (February), 29 (February leap year), 30, or 31. First day of the month is 1. |

### 7.1.42 RTC Set Time of Day (RTC_SET_TIME)

Offset: 0x0C0000B4
Reset Value: See field descriptions
Access: Read/Write

To set the RTC time of day, software writes the target time to the RTC_SET_TIME and RTC_SET_DATE registers, then writes a 1 to the LOAD_RTC bit of the RTC_CONTROL register. The LOAD_RTC bit also sets the sub-second counter to 0.

This register is only reset by RTC_RESET.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31:27 | RES | 0x0 | Reserved |
| 26:24 | WEEK_DAY | 0x1 | Set the weekday, which changes at midnight; number rolls after 7 (1=Sunday, 2=Monday, etc.). |
| 23:22 | RES | 0x0 | Reserved |
| 21:16 | HOUR | 0x0 | Set the hour: 0–23 in 24-hour mode or 0–12 AM/PM in twelve-hour mode. Resets to midnight, BCD.<br>This setting must be consistent with the programming of BCD. |
| 15 | RES | 0x0 | Reserved |
| 14:8 | MINUTE | 0x0 | Set the minute count; rolls after 59 |
| 7 | RES | 0x0 | Reserved |
| 6:0 | SECOND | 0x0 | Set the second count; rolls after 59 |

### 7.1.43 RTC Set Date and Year (RTC_SET_DATE)

Offset: 0x0C0000B8
Reset Value: See field descriptions
Access: Read/Write

To set the RTC date and year, software writes the target time to the RTC_SET_TIME and RTC_SET_DATE registers, then writes a 1 to the LOAD_RTC bit of the RTC_CONTROL register.

This register is only reset by RTC_RESET.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31:24 | RES | 0x0 | Reserved |
| 23:16 | YEAR | 0x0 | Set the current year (0 = 20000) |
| 15:13 | RES | 0x0 | Reserved |
| 12:8 | MONTH | 0x1 | Set the current month count; rolls after 12 (1=January, 2=February, etc.) |
| 7:6 | RES | 0x0 | Reserved |
| 5:0 | MONTH_DAY | 0x1 | Set the current day of the month, rolls at 28 (February), 29 (February leap year), 30, or 31. First day of the month is 1. |

### *7.1.44 RTC Alarm Time of Day (RTC_SET_ALARM)*

Offset: 0x0C0000BC
Reset Value: 0x0
Access: Read/Write

Returns current alarm time values on read and updates alarm time on write.

The alarm time is only loaded into the RTC logic after software writes a 1 to the LOAD_ALARM bit in the RTC_CONTROL register. If the time is set to an illegal value, the alarm never triggers (e.g., if SECOND = 61).

This register is only reset by RTC_RESET.

| Bits | Bit Name | Description |
|---|---|---|
| 31:22 | RES | Reserved |
| 21:16 | HOUR | Set the alarm hour: 0–23 in 24-hour mode or 0–12 AM/PM in twelve-hour mode. Resets to midnight, BCD.<br>This setting must be consistent with the programming of BCD. |
| 15 | RES | Reserved |
| 14:8 | MINUTE | Set the alarm minute count; rolls after 59 |
| 7 | RES | Reserved |
| 6:0 | SECOND | Set the alarm second count; rolls after 59 |

### *7.1.45 RTC Operation Configuration (RTC_CONFIG)*

Offset: 0x0C0000C0
Reset Value: See field descriptions
Access: Read/Write

Read or write RTC configuration options.

When software changes the value of any of the fields in this register, it should also update the RTC_SET_TIME and RTC_SET_DATE registers and write a 1 to the LOAD_RTC bit in the RTC_CONTROL register, to push the configuration to the RTC logic.

This register is only reset by RTC_RESET.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31:3 | RES | 0x0 | Reserved |
| 2 | BCD | 0x1 | When set to 1, the RTC operates binary coded decimal.<br>In BCD mode, all writes to the RTC_SET_TIME, RTC_SET_DATE, and RTC_SET_ALARM_TIME should be formatted in BCD. All reads from the RTC_TIME and RTC_DATE registers return BCD values when the BCD bit is set.<br>When this field is set to 0, the RTC operates in binary mode. Accesses to the RTC_SET_TIME, RTC_SET_DATE, and RTC_SET_ALARM_TIME are formatted in ordinary binary. |

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 1 | TWELVE_HOUR | 0x1 | The RTC must be in BCD mode for 12-hour mode to engage. When set to 1, and when the BCD bit is also set to 1, the RTC operates in 12-hour mode. In 12-hour mode, the 8-bit hour fields mean:<br>■ hour[7:6] Always 0<br>■ hour[5] = AM/PM (AM=0)<br>■ hour[4] = Tens hour<br>■ hour[3:0] Ones hour<br>When set to 0, the RTC operates in 24-hour mode, with the bit decode:<br>■ hour[7:6] Always 0<br>■ hour[5:4] Tens hour<br>■ hour[3:0] Ones hour<br>In 12-hour mode, all writes to SET_ALARM_TIME and SET_RTC_TIME should use 12-hour format. All reads to RTC_TIME return in 12-hour format when the TWELVE_HOUR bit is set. |
| 0 | DSE | 0x1 | When set to 1, daylight savings updates are enabled:<br>■ First Sunday of April, time changes from 01:59:59 to 03:00:00.<br>■ Last Sunday of October, time changes from 01:59:59 to 01:00:00.<br>When this field is set to 0, no special updates occur. |

### *7.1.46 RTC Alarm Enable, Set and Clear (RTC_ALARM_STATUS)*

Read or write the RTC alarm enable and interrupt bit. This register is only reset by RTC_RESET.

Offset: 0x0C0000C4
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | ENABLE | This field enables the RTC_ALARM interrupt. If the alarm is disabled, it never wakes the phone from OFF or SLEEP state. This signal synchronizes to the RTC clock before it is sampled, so a ~45 µs delay exists between changing this signal and its taking effect.<br>■ 1 = RTC Alarm is enabled<br>■ 0 = RTC Alarm is disabled |
| 0 | INTERRUPT | If the RTC hour, minute, and second values match the values set in the RTC_ALARM_TIME register, this bit will be set by hardware. Software can also set this bit with a write. When set by either hardware or software, the RTC_ALARM source interrupt signal will be asserted to the interrupt controller. |

### *7.1.47 UART Wakeup Events Enable (UART_WAKEUP)*

This register enables UART wakeup events (activity on the Rx line) to wake the system out of sleep.

Offset: 0x0C0000C8
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:1 | | Reserved |
| 0 | ENABLE | ■ 0 = UART Rx transitions do not cause a wakeup event<br>■ 1 = UART Rx transitions causes a wakeup event |

### *7.1.48 Reset Cause (RESET_CAUSE)*

Reset Value: 0x0
Access: Read only

Offset: 0x0C0000CC

This register holds the cause of the last reset event, allowing software to detect watchdog reset events and other initial conditions.

| Bits | Bit Name | Description |
|---|---|---|
| 31:3 | RES | Reserved |
| 2:0 | LAST | The value of this register hold the last cause of RESET.<br>■ 0 = The SYS_RST_L pin was asserted<br>■ 1 = The host wrote to the SDIO reset register<br>■ 2 = Software wrote RTC_CONTROL_COLD_RST register<br>■ 3 = Software wrote RTC_CONTROL_WARM_RST register<br>■ 4 = Software wrote RTC_CONTROL_CPU_RST register<br>■ 5 = The watchdog timer expired<br>■ 6–7 = Reserved |

### *7.1.49 System Sleep Status (SYSTEM_SLEEP)*

Offset: 0x0C0000D0
Reset Value: See field descriptions
Access: Read only

System sleep state is entered when all high frequency clocks are gated and the high frequency crystal is shut down. This register indicates the status of each sleep control interface. If any bit in this control register is 0, sleep is not permitted. If all bits are 1, sleep is permitted. The system enters sleep as soon as the CPU executes a WAIT instruction.

The LIGHT field gates clocks off in SLEEP, but keeps the crystal running for faster wakeup.

The DISABLE field prevents the chip from entering SLEEP.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31:5 | | 0x0 | Reserved |
| 4 | HOST_IF | 0x1 | ■ 0 = The host interface does not allow sleep state<br>■ 1 = The host interface has enabled the sleep state |
| 3 | MBOX | 0x1 | ■ 0 = The Mbox interface does not allow sleep state<br>■ 1 = The Mbox interface has enabled the sleep state |
| 2 | MAC_IF | 0x1 | ■ 0 = The MAC block does not allow sleep state<br>■ 1 = The MAC block has enabled the sleep state |
| 1 | LIGHT | 0x0 | Controls whether or not the crystal is turned off during SLEEP. If the crystal is turned off, power is less during sleep but the wakeup time is XTAL_SETTLE. If the crystal us left on, power consumption is higher but the wakeup time is ~45 µs.<br>■ 0 = System sleep DEEP, minimal power consumption<br>■ 1 = System sleep LIGHT |
| 0 | DISABLE | 0x0 | ■ 1 = System sleep disabled<br>■ 0 = System sleep enabled |

### *7.1.50 LDO_D Voltage (LDO_VOLTAGE)*

Offset: 0x0C0000D4
Reset Value: 0x0
Access: Read/Write

This register selects LDO_D voltage in various modes.

Encoding for voltage levels are:

- 000 = 1.8V
- 111 = 1.7V
- 110 = 1.6V
- 101 = 1.5V
- 100 = 1.4V
- 011 = 1.3V
- 010 = 1.2V
- 001 = 1.1V

| Bits | Bit Name | Description |
|---|---|---|
| 31:15 | RES | Reserved |
| 14:12 | SLEEP | LDO voltage during SLEEP state. No high speed clock runs in this state. |
| 11 | RES | Reserved |
| 10:8 | WAKEUP | LDO voltage during WAKEUP state. Occurs while the XTAL is warming up. High speed clocks are gated off. |
| 7 | RES | Reserved |
| 6:4 | SOC_ON | LDO voltage during SOC_ON state when the CPU and/or MBOX blocks are active, but the MAC, BB, and Radio are asleep (clocks gated off). |
| 3 | RES | Reserved |
| 2:0 | ON | LDO voltage during ON state |

### *7.1.51 LDO_A Voltage (LDO_A_VOLTAGE)*

Offset: 0x0C0000D8
Reset Value: 0x0
Access: Read/Write

This register selects the LDO_A voltage in various modes.

Encoding for voltage levels are as follows:

- 000 = 1.8V
- 111 = 1.7V
- 110 = 1.6V
- 101 = 1.5V
- 100 = 1.4V
- 011 = 1.3V
- 010 = 1.2V
- 001 = 1.1V

| Bits | Bit Name | Description |
|---|---|---|
| 31:15 | RES | Reserved |
| 14:12 | SLEEP | LDO voltage during SLEEP state. No high speed clock runs in this state. |
| 11 | RES | Reserved |
| 10:8 | WAKEUP | LDO voltage during WAKEUP state. Occurs while the XTAL is warming up. High speed clocks are gated off. |
| 7 | RES | Reserved |
| 6:4 | SOC_ON | LDO voltage during SOC_ON state when the CPU and/or MBOX blocks are active, but the MAC, BB, and Radio are asleep (clocks gated off). |
| 3 | RES | Reserved |
| 2:0 | ON | LDO voltage during ON state |

### *7.1.52 SDIO_LDO voltage (SDIO_LDO_VOLTAGE)*

Offset: 0x0C0000DC
Reset Value: 0x0
Access: Read/Write

This register selects the SDIO_LDO voltage in various modes.

Encoding for voltage levels are as follows:

- 000 = 1.8V
- 111 = 1.7V
- 110 = 1.6V
- 101 = 1.5V
- 100 = 1.4V
- 011 = 1.3V
- 010 = 1.2V
- 001 = 1.1V

| Bits | Bit Name | Description |
|---|---|---|
| 31:19 | RES | Reserved |
| 18:16 | OFF | SDIO_LDO voltage setting when SDIO CCCR enable bit disables the AR6001X |
| 15 | RES | Reserved |
| 14:12 | SLEEP | LDO voltage during SLEEP state. No high speed clock runs in this state. |
| 11 | RES | Reserved |
| 10:8 | WAKEUP | LDO voltage during WAKEUP state. Occurs while the XTAL is warming up. High speed clocks are gated off. |
| 7 | RES | Reserved |
| 6:4 | SOC_ON | LDO voltage during SOC_ON state when the CPU and/or MBOX blocks are active, but the MAC, BB, and Radio are asleep (clocks gated off). |
| 3 | RES | Reserved |
| 2:0 | ON | LDO voltage during ON state |

### 7.1.53 Core Pad Enable (CORE_PAD_ENABLE)

Offset: 0x0C0000E0
Reset Value: 0x1
Access: Read/Write

This register controls the core pad enable. The CORE PADS must turn off when internal voltage is too low (i.e., LDO_D has been set too low) for the pads to operate properly. This register controls the pad enable for each state.

| Bits | Bit Name | Description |
|---|---|---|
| 31:4 | RES | Reserved |
| 3 | SLEEP | Core pad enable state during SLEEP. No high speed clock runs in this state. |
| 2 | WAKEUP | Core pad enable state during WAKEUP. Occurs while the XTAL is warming up. High speed clocks are gated off. |
| 1 | SOC_ON | Core pad enable state during SOC_ON when the CPU and/or MBOX blocks are active, but the MAC, BB, and Radio are asleep (clocks gated off). |
| 0 | ON | Core pad enable state during ON |

### 7.1.54 SDIO Signal Wrapper (SDIO_WRAPPER)

Offset: 0x0C0000E4
Reset Value: 0x1
Access: Read/Write

This register controls the SDIO signal wrapper. When the wrapper is enabled, or set to 1, signals pass from the core to the SDIO block. When the wrapper is disabled, or set to 0, all signals from core to SDIO blocks are gated off to 0.

| Bits | Bit Name | Description |
|---|---|---|
| 31:4 | RES | Reserved |
| 3 | SLEEP | Signal wrapper state during SLEEP. No high speed clock runs in this state. |
| 2 | WAKEUP | Signal wrapper state during WAKEUP. Occurs while the XTAL is warming up. High speed clocks are gated off. |
| 1 | SOC_ON | Signal wrapper state during SOC_ON when the CPU and/or MBOX blocks are active, but the MAC, BB, and Radio are asleep (clocks gated off). |
| 0 | ON | Signal wrapper state during ON |

### 7.1.55 MAC Sleep Options (MAC_SLEEP_CONTROL)

Offset: 0x0C0000E8
Reset Value: 0x2
Access: Read/Write

This register controls MAC sleep options. The MAC can be forced awake or asleep by the CPU. The force options are not intended for operation, but only as a test feature.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1:0 | ENABLE | After a force awake or asleep event, the MAC behavior is not defined. To exit a FORCE state, this register should be set to NORMAL to reset the MAC.<br>Encoding:<br>■ 0 = Force MAC wake<br>■ 1 = Force MAC sleep<br>■ 2 = Normal (MAC controlled) sleep<br>■ 3 = reserved |

### 7.1.56 Keep Awake Timer (KEEP_AWAKE)

Offset: 0x0C0000EC
Reset Value: 0x2
Access: Read/Write

The keep awake timer ensures that the chip does not enter SLEEP until at least COUNT cycles have passed from the time of the last CLK_REQ event.

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7:0 | COUNT | Keep awake timer measured in 32 KHz (30.5 µs) cycles |

### 7.1.57 Chip Rev ID (CHIP_REV)

Offset: 0x0C0000F0
Reset Value: 0x0
Access: Read only

This register returns the chip revision.

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7:0 | ID | The revision ID is set at chip fabrication |

### 7.1.58 HF 32 KHz Clock Creation (DERIVED_RTC_CLK)

Offset: 0x0C0000F4
Reset Value: 0x0
Access: Read/Write

This register controls a scaled output clock that can generate lower frequency clocks based on the reference clock. For example, it can generate a 32768 KHz clock by setting the divisor of the high speed clock accordingly. The accuracy depends on how divisors align to this integer count. The AR6001X always boots using the derived RTC_CLK and switches to LF_XTAL if it detects an LF_XTAL. This register can override LF_XTAL selection.

| Bits | Bit Name | Description |
|---|---|---|
| 31:19 | RES | Reserved |
| 18 | EXTERNAL_DETECT | Detects an external 32KHz XTAL. If a LF XTAL is detected and the FORCE field is cleared, AR6001 will automatically use the external XTAL.<br>■ 1 = XTAL detected on LFXTAL<br>■ 0 = No XTAL detected |
| 17:16 | FORCE | Forces use of the derived out clock logic.<br>■ 0x = No force. Allow hardware to detect LF_XTAL and use it as the sleep clock if found.<br>■ 10 = Force the sleep clock to source from LF_XTAL<br>■ 11 = Force the sleep clock to source from the derived clock |
| 15:1 | PERIOD | The period of the derived out clock. This field actually counts half the period. The default value creates a 32768 Hz clock if the refClk is 32 KHz. |
| 0 | RES | Reserved |

### 7.1.59 Automatic Clock Gating Control (ACG_DISABLE)

Offset: 0x0C0000F8
Reset Value: 0x0
Access: Read/Write

This register controls the automatic clock gating (ACG) circuits in the AR6001X. The ACG circuits are enabled by default as they save power. They can be disabled for testing purposes.

| Bits | Bit Name | Description |
|---|---|---|
| 31:4 | RES | Reserved |
| 3 | CPU | Disables ACG for the CPU block.<br>■ 1 = ACG disabled<br>■ 0 = ACG enabled |
| 2 | SDIO | Disables ACG for the SDIO block.<br>■ 1 = ACG disabled<br>■ 0 = ACG enabled |
| 1 | BB_AND_BBB | Disables ACG for the BB_AND_BBB block.<br>■ 1 = ACG disabled<br>■ 0 = ACG enabled |
| 0 | AMBA_MAC | Disables ACG for the AMBA_MAC block.<br>■ 1 = ACG disabled<br>■ 0 = ACG enabled |

## 7.2 Memory Block Registers

Table 7-2 summarizes memory block registers.

*Table 7-2.* **Memory Block Registers**

| Offset | Name | Description | Page |
|---|---|---|---|
| 0x0C004000 | BANK0_ADDR | Bank 0 address | page 77 |
| 0x0C004004 | BANK0_CONFIG | Bank 0 configuration | page 78 |
| 0x0C004008 | BANK0_READ | Bank 0 read sequence | page 79 |
| 0x0C00400C | BANK0_WRITE | Bank 0 write sequence | page 80 |
| 0x0C004010 | BANK1_ADDR | Bank 1 address | page 81 |
| 0x0C004014 | BANK1_CONFIG | Bank 1 configuration | page 81 |
| 0x0C004018 | BANK1_READ | Bank 1 read sequence | page 82 |
| 0x0C00401C | BANK1_WRITE | Bank 1 write sequence | page 83 |
| 0x0C004020 | BANK2_ADDR | Bank 2 address | page 84 |
| 0x0C004024 | BANK2_CONFIG | Bank 2 configuration | page 84 |
| 0x0C004028 | BANK2_READ | Bank 2 read sequence | page 85 |
| 0x0C00402C | BANK2_WRITE | Bank 2 write sequence | page 86 |
| 0x0C004344 | TIMING_INT_ENABLE | Timing interrupt enable | page 87 |
| 0x0C004348 | MC_ERROR_STATUS | MC interrupt status | page 87 |

### 7.2.1 Bank 0 Address (BANK0_ADDR)

Offset: 0x0C004000
Reset Value: See field descriptions
Access: Read/Write

This register specifies the base address and size of the Bank 0 Address Space. Within the 256 MB address space, addresses from 32 MB to 64 MB are allocated for internal registers, and accesses to that range never reaches the memory controller. Each of the three memory banks can be from 1 KB to 32 MB in size and be allocated in any part of the 0–32 MB and the 64–256 MB address ranges. If two or more banks are allocated to the same address range, neither bank is accessed and an error response is generated. Memory banks can only be allocated at addresses that are an exact multiple of their size. For example, a 256 KB memory bank could be allocated at address 0, 256 KB, 51 2KB, etc. Reset values are set up so all banks default to 32 MB and are located consecutively starting at 64 MB.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31:28 | SIZE | 0xF | The size of Bank 0 is represented in this field as $2^{SIZE}$ KB |
| 27:10 | BASE | 0x8000 | This field contains bits 27:10 of Bank 0's base address. Only address bits larger than the bank size are used. For example, if the SIZE field is set to 0x8 or 256 KB, then {BASE[27:18], 18'h0} is used as the base address and BASE[17:10] is ignored. |
| 9:0 | RES | 0x0 | Reserved |

### *7.2.2 Bank 0 Configuration (BANK0_CONFIG)*

Offset: 0x0C004004
Reset Value: See field descriptions
Access: Read/Write

This register sets basic parameters for read and write accesses to Bank 0. The upper half of the register sets configuration parameters such as bank enable, bank width, write protect, and write buffer control. The lower half sets the timer values used to sequence both read and write accesses. Bank 0 (flash) defaults to being enabled; all other banks default to disabled. All banks default to 8 bits wide.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31 | ENABLE | 0x1 | Enables Bank 0 for use |
| 30:29 | RES | 0x0 | Reserved |
| 28 | WIDTH | 0x0 | Sets the width of the data bus for this part (1 = 16bit, 0 = 8bit) |
| 27 | RES | 0x0 | Reserved |
| 26 | PROTECT | 0x0 | Prevents any write to Bank 0 from updating memory and forces all writes to return an ERROR response |
| 25 | WB_ENABLE | 0x0 | If this bit is set, writes are buffered/posted and held off until an address match, a buffer conflict, or an explicit flush command forces the data to be written to memory. If this bit is not set, write data is sent to memory as soon as it is available. |
| 24 | WB_FLUSH | 0x0 | Forces a flush of the write buffers to memory; hardware clears it once the flush completes |
| 23:22 | RES | 0x0 | Reserved |
| 21:20 | SCALE | 0x3 | Set the number of clock cycles between any increments of the timers: the scale is logarithmic (number of cycles = $2^{SCALE}$) |
| 19:16 | HOLDOFF | 0xF | Sets the number of clock cycles for the memory controller to wait between any two non-burst accesses to Bank 0 |
| 15:12 | TIMER3 | 0xF | Sets the number of clock cycles between event 3 and event 4 for any Bank 0 access |
| 11:8 | TIMER2 | 0xF | Sets the number of clock cycles between event 2 and event 3 for any Bank 0 access |
| 7:4 | TIMER1 | 0xF | Sets the number of clock cycles between event 1 and event 2 for any Bank 0 access |
| 3:0 | TIMER0 | 0xF | Sets the number of clock cycles between event 0 and event 1 for any Bank 0 access |

### 7.2.3 Bank 0 Read Sequence (BANK0_READ)

Offset: 0x0C004008
Reset Value: See field descriptions
Access: Read/Write

This register controls the sequence of events needed to execute a read of the device mapped to Bank 0. The default sequence is a standard SRAM-style access. CS is asserted on event 0; OE is asserted on event 1; OE is deasserted and data is captured on event 2; and CS is deasserted on event 3. In any sequence, the address is valid starting at event 0 and ending at either the END_EVENT or the BURST_END_EVENT.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31 | ENABLE_WAIT | 0x0 | If set, extends one of the four periods until the WAIT signal has been deasserted. The period to extend is selected by the WAIT_EVENT field in this register. |
| 30:28 | WAIT_EVENT | 0x2 | Indicates which event should delay based on the WAIT signal. Only used if the ENABLE_WAIT bit in this register is set. |
| 27 | RES | 0x0 | Reserved |
| 26:24 | END_EVENT | 0x3 | Indicates the final event in the read sequence. When this event is reached, all control signals return to their default states and the HOLDOFF counter starts. |
| 23 | RES | 0x0 | Reserved |
| 22:20 | BURST_END_EVENT | 0x7 | Indicates the final event in a burst read sequence. When this event is reached in the middle of a burst, the read sequence immediately jumps to the BURST_START_EVENT. If the value in this register is 0x7, bursting is disabled and consecutive burst requests complete as individual transactions (complete with holdoff). |
| 19 | RES | 0x0 | Reserved |
| 18:16 | BURST_START_EVENT | 0x7 | Indicates the start event of a burst read sequence. If the value in this register is 0x7, bursting is disabled and consecutive burst requests complete as individual transactions (complete with holdoff). |
| 15 | EVENT3_DC | 0x0 | If Set, the read data is captured on the same cycle as event 3 |
| 14 | EVENT3_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 3 and event 4 |
| 13 | EVENT3_OE | 0x0 | If set, the OE signal is set between event 3 and event 4. |
| 12 | EVENT3_CS | 0x0 | If set, the Bank 0 CS signal is set between event 3 and event 4. |
| 11 | EVENT2_DC | 0x1 | If Set, the read data is captured on the same cycle as event 2 |
| 10 | EVENT2_BE | 0x0 | If set, the BE signals (UBE/LBE) are set appropriately between event 2 and event 3 |
| 9 | EVENT2_OE | 0x0 | If set, the OE signal is set between event 2 and event 3. |
| 8 | EVENT2_CS | 0x1 | If set, the Bank 0 CS signal is set between event 2 and event 3 |
| 7 | EVENT1_DC | 0x0 | If set, the read data is captured on the same cycle as event 1 |
| 6 | EVENT1_BE | 0x0 | If set, the BE signals (UBE/LBE) are set appropriately between event 1 and event 2 |
| 5 | EVENT1_OE | 0x1 | If set, the OE signal is set between event 1 and event 2 |
| 4 | EVENT1_CS | 0x1 | If set, the Bank 0 CS signal is set between event 1 and event 2 |
| 3 | EVENT0_DC | 0x0 | If Set, the read data is captured on the same cycle as event 0 |
| 2 | EVENT0_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 0 and event 1 |
| 1 | EVENT0_OE | 0x0 | If set, the OE signal is set between event 0 and event 1 |
| 0 | EVENT0_CS | 0x1 | If set, the Bank 0 CS signal is set between event 0 and event 1 |

### 7.2.4 Bank 0 Write Sequence (BANK0_WRITE)

Offset: 0x0C00400C
Reset Value: See field descriptions
Access: Read/Write

This register controls the sequence of events needed to execute a write of the device mapped to Bank 0. In any transaction, the address and write data are valid starting at event 0 and ending at either the END_EVENT or the BURST_END_EVENT.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31 | ENABLE_WAIT | 0x0 | If set, extend one of the four periods until the WAIT signal deasserts. The WAIT_event field in this register selects the period to extend. |
| 30:28 | WAIT_EVENT | 0x2 | Indicates which event should be delayed based on the WAIT signal; only used if the ENABLE_WAIT bit in this register is set. |
| 27 | RES | 0x0 | Reserved |
| 26:24 | END_EVENT | 0x3 | Indicates the final event in the write sequence. When this event is reached, control signals return to their default states and the HOLDOFF counter starts. |
| 23 | RES | 0x0 | Reserved |
| 22:20 | BURST_END_EVENT | 0x7 | Indicates the final event in a burst write sequence. When this event is reached in the middle of a burst, the write sequence immediately jumps to the BURST_START_event. If the value in this register is 0x7, bursting is disabled and consecutive burst requests complete as individual transactions (complete with holdoff). |
| 19 | RES | 0x0 | Reserved |
| 18:16 | BURST_START_EVENT | 0x7 | Indicates the start event of a burst write sequence. If the value in this register is 0x7, bursting is disabled and consecutive burst requests complete as individual transactions (complete with holdoff). |
| 15 | RES | 0x0 | Reserved |
| 14 | EVENT3_BE | 0x0 | If set, the BE signals (UBE/LBE) are set appropriately between event 3 and event 4 |
| 13 | EVENT3_WE | 0x0 | If set, the WE signal is set between event 3 and event 4 |
| 12 | EVENT3_CS | 0x0 | If set, the Bank 0 CS signal is set between event 3 and event 4 |
| 11 | | 0x0 | Reserved |
| 10 | EVENT2_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 2 and event 3 |
| 9 | EVENT2_WE | 0x0 | If set, the WE signal is set between event 2 and event 3 |
| 8 | EVENT2_CS | 0x1 | If set, the Bank 0 CS signal is set between event 2 and event 3 |
| 7 | | 0x0 | Reserved |
| 6 | EVENT1_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 1 and event 2 |
| 5 | EVENT1_WE | 0x1 | If set, the WE signal is set between event 1 and event 2 |
| 4 | EVENT1_CS | 0x1 | If set, the Bank 0 CS signal is set between event 1 and event 2 |
| 3 | | 0x0 | Reserved |
| 2 | EVENT0_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 0 and event 1 |
| 1 | EVENT0_WE | 0x0 | If set, the WE signal is set between event 0 and event 1 |
| 0 | EVENT0_CS | 0x1 | If set, the Bank 0 CS signal is set between event 0 and event 1 |

### 7.2.5 Bank 1 Address (BANK1_ADDR)

Offset: 0x0C004010
Reset Value: See field descriptions
Access: Read/Write

This register specifies the base address and size of the Bank 1 address space.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31:28 | SIZE | 0xF | The size of Bank 1 is represented in this field as ($2^{SIZE}$) KB |
| 27:10 | BASE | 0x10000 | This field contains bits 27:10 of Bank 1's base address. Only those address bits that are larger than the bank size are used. For example, if the SIZE field is set to 0x8 or 256 KB, then {BASE[27:18], 18'h0} is used as the base address and BASE[17:10] is ignored. |
| 9:0 | RES | 0x0 | Reserved |

### 7.2.6 Bank 1 Configuration (BANK1_CONFIG)

Offset: 0x0C004014
Reset Value: See field descriptions
Access: Read/Write

This register sets basic parameters for read and write accesses to Bank 1. The upper half of the register sets configuration parameters such as bank enable, bank width, write protect, and write buffer control. The lower half sets the timer values used to sequence both read and write accesses. Bank 1 (flash) defaults to being enabled; all other banks default to disabled. All banks default to 8 bits wide.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31 | ENABLE | 0x1 | Enables Bank 1 for use |
| 30:29 | RES | 0x0 | Reserved |
| 28 | WIDTH | 0x0 | Sets the width of the data bus for this part (1 = 16bit, 0 = 8bit) |
| 27 | RES | 0x0 | Reserved |
| 26 | PROTECT | 0x0 | Prevents any write to Bank 1 from updating memory and forces all writes to return an ERROR response |
| 25 | WB_ENABLE | 0x0 | If this bit is set, writes are buffered/posted and held off until an address match, a buffer conflict, or an explicit flush command forces the data to be written to memory. If this bit is not set, write data is sent to memory as soon as it is available. |
| 24 | WB_FLUSH | 0x0 | Forces a flush of the write buffers to memory; hardware clears it once the flush completes |
| 23:22 | RES | 0x0 | Reserved |
| 21:20 | SCALE | 0x3 | Set the number of clock cycles between any increments of the timers. The scale is logarithmic (number of cycles = $2^{SCALE}$). |
| 19:16 | HOLDOFF | 0xF | Sets the number of clock cycles for the memory controller to wait between any two non-burst accesses to Bank 1 |
| 15:12 | TIMER3 | 0xF | Sets the number of clock cycles between event 3 and event 4 for any Bank 1 access |
| 11:8 | TIMER2 | 0xF | Sets the number of clock cycles between event 2 and event 3 for any Bank 1 access |
| 7:4 | TIMER1 | 0xF | Sets the number of clock cycles between event 1 and event 2 for any Bank 1 access |
| 3:0 | TIMER0 | 0xF | Sets the number of clock cycles between event 0 and event 1 for any Bank 1 access |

### *7.2.7 Bank 1 Read Sequence (BANK1_READ)*

Offset: 0x0C004018
Reset Value: See field descriptions
Access: Read/Write

This register controls the sequence of events needed to execute a read of the device mapped to Bank 0. The default sequence is a standard SRAM-style access. CS is asserted on event 0; OE is asserted on event 1; OE is deasserted and data is captured on event 2; and CS is deasserted on event 3. In any sequence, the address is valid starting at event 0 and ending at either the END_EVENT or the BURST_END_EVENT.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31 | ENABLE_WAIT | 0x0 | If set, extends one of the four periods until the WAIT signal has been deasserted. The period to extend is selected by the WAIT_EVENT field in this register. |
| 30:28 | WAIT_EVENT | 0x2 | Indicates which event should delay based on the WAIT signal. Only used if the ENABLE_WAIT bit in this register is set. |
| 27 | RES | 0x0 | Reserved |
| 26:24 | END_EVENT | 0x3 | Indicates the final event in the read sequence. When this event is reached, all control signals return to their default states and the HOLDOFF counter starts. |
| 23 | RES | 0x0 | Reserved |
| 22:20 | BURST_END_EVENT | 0x7 | Indicates the final event in a burst read sequence. When this event is reached in the middle of a burst, the read sequence immediately jumps to the BURST_START_EVENT. If the value in this register is 0x7, bursting is disabled and consecutive burst requests complete as individual transactions (complete with holdoff). |
| 19 | RES | 0x0 | Reserved |
| 18:16 | BURST_START_EVENT | 0x7 | Indicates the start event of a burst read sequence. If the value in this register is 0x7, bursting is disabled and consecutive burst requests complete as individual transactions (complete with holdoff). |
| 15 | EVENT3_DC | 0x0 | If Set, the read data is captured on the same cycle as event 3 |
| 14 | EVENT3_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 3 and event 4 |
| 13 | EVENT3_OE | 0x0 | If set, the OE signal is set between event 3 and event 4. |
| 12 | EVENT3_CS | 0x0 | If set, the Bank 0 CS signal is set between event 3 and event 4. |
| 11 | EVENT2_DC | 0x1 | If Set, the read data is captured on the same cycle as event 2 |
| 10 | EVENT2_BE | 0x0 | If set, the BE signals (UBE/LBE) are set appropriately between event 2 and event 3 |
| 9 | EVENT2_OE | 0x0 | If set, the OE signal is set between event 2 and event 3. |
| 8 | EVENT2_CS | 0x1 | If set, the Bank 0 CS signal is set between event 2 and event 3 |
| 7 | EVENT1_DC | 0x0 | If set, the read data is captured on the same cycle as event 1 |
| 6 | EVENT1_BE | 0x0 | If set, the BE signals (UBE/LBE) are set appropriately between event 1 and event 2 |
| 5 | EVENT1_OE | 0x1 | If set, the OE signal is set between event 1 and event 2 |
| 4 | EVENT1_CS | 0x1 | If set, the Bank 0 CS signal is set between event 1 and event 2 |
| 3 | EVENT0_DC | 0x0 | If Set, the read data is captured on the same cycle as event 0 |
| 2 | EVENT0_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 0 and event 1 |
| 1 | EVENT0_OE | 0x0 | If set, the OE signal is set between event 0 and event 1 |
| 0 | EVENT0_CS | 0x1 | If set, the Bank 0 CS signal is set between event 0 and event 1 |

### 7.2.8 Bank 1 Write Sequence (BANK1_WRITE)

Offset: 0x0C00401C
Reset Value: See field descriptions
Access: Read/Write

This register controls the sequence of events needed to execute a write of the device mapped to Bank 1. In any transaction, the address and write data are valid starting at event 0 and ending at either the END_EVENT or the BURST_END_EVENT.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31 | ENABLE_WAIT | 0x0 | If set, extend one of the four periods until the WAIT signal deasserts. The WAIT_event field in this register selects the period to extend. |
| 30:28 | WAIT_EVENT | 0x2 | Indicates which event should be delayed based on the WAIT signal; only used if the ENABLE_WAIT bit in this register is set. |
| 27 | RES | 0x0 | Reserved |
| 26:24 | END_EVENT | 0x3 | Indicates the final event in the write sequence. When this event is reached, control signals return to their default states and the HOLDOFF counter starts. |
| 23 | RES | 0x0 | Reserved |
| 22:20 | BURST_END_EVENT | 0x7 | Indicates the final event in a burst write sequence. When this event is reached in the middle of a burst, the write sequence immediately jumps to the BURST_START_event. If the value in this register is 0x7, bursting is disabled and consecutive burst requests complete as individual transactions (complete with holdoff). |
| 19 | RES | 0x0 | Reserved |
| 18:16 | BURST_START_EVENT | 0x7 | Indicates the start event of a burst write sequence. If the value in this register is 0x7, bursting is disabled and consecutive burst requests complete as individual transactions (complete with holdoff). |
| 15 | RES | 0x0 | Reserved |
| 14 | EVENT3_BE | 0x0 | If set, the BE signals (UBE/LBE) are set appropriately between event 3 and event 4 |
| 13 | EVENT3_WE | 0x0 | If set, the WE signal is set between event 3 and event 4 |
| 12 | EVENT3_CS | 0x0 | If set, the Bank 1 CS signal is set between event 3 and event 4 |
| 11 | | 0x0 | Reserved |
| 10 | EVENT2_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 2 and event 3 |
| 9 | EVENT2_WE | 0x0 | If set, the WE signal is set between event 2 and event 3 |
| 8 | EVENT2_CS | 0x1 | If set, the Bank 1 CS signal is set between event 2 and event 3 |
| 7 | | 0x0 | Reserved |
| 6 | EVENT1_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 1 and event 2 |
| 5 | EVENT1_WE | 0x1 | If set, the WE signal is set between event 1 and event 2 |
| 4 | EVENT1_CS | 0x1 | If set, the Bank 1 CS signal is set between event 1 and event 2 |
| 3 | | 0x0 | Reserved |
| 2 | EVENT0_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 0 and event 1 |
| 1 | EVENT0_WE | 0x0 | If set, the WE signal is set between event 0 and event 1 |
| 0 | EVENT0_CS | 0x1 | If set, the Bank 1 CS signal is set between event 0 and event 1 |

### 7.2.9 Bank 2 Address (BANK2_ADDR)

Offset: 0x0C004020
Reset Value: See field descriptions
Access: Read/Write

This register specifies the base address and size of the Bank 2 address space.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31:28 | SIZE | 0xF | The size of Bank 2 is represented in this field as ($2^{SIZE}$) KB |
| 27:10 | BASE | 0x10000 | This field contains bits 27:10 of Bank 2's base address. Only those address bits that are larger than the bank size are used. For example, if the SIZE field is set to 0x8 or 256 KB, then {BASE[27:18], 18'h0} is used as the base address and BASE[17:10] is ignored. |
| 9:0 | RES | 0x0 | Reserved |

### 7.2.10 Bank 2 Configuration (BANK2_CONFIG)

Offset: 0x0C004024
Reset Value: See field descriptions
Access: Read/Write

This register sets basic parameters for read and write accesses to Bank 2. The upper half of the register sets configuration parameters such as bank enable, bank width, write protect, and write buffer control. The lower half sets the timer values used to sequence both read and write accesses. Bank 2 (flash) defaults to being enabled; all other banks default to disabled. All banks default to 8 bits wide.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31 | ENABLE | 0x1 | Enables Bank 2 for use |
| 30:29 | RES | 0x0 | Reserved |
| 28 | WIDTH | 0x0 | Sets the width of the data bus for this part (1 = 16bit, 0 = 8bit) |
| 27 | RES | 0x0 | Reserved |
| 26 | PROTECT | 0x0 | Prevents any write to Bank 2 from updating memory and forces all writes to return an ERROR response |
| 25 | WB_ENABLE | 0x0 | If this bit is set, writes are buffered/posted and held off until an address match, a buffer conflict, or an explicit flush command forces the data to be written to memory. If this bit is not set, write data is sent to memory as soon as it is available. |
| 24 | WB_FLUSH | 0x0 | Forces a flush of the write buffers to memory; hardware clears it once the flush completes |
| 23:22 | RES | 0x0 | Reserved |
| 21:20 | SCALE | 0x3 | Set the number of clock cycles between any increments of the timers. The scale is logarithmic (number of cycles = $2^{SCALE}$). |
| 19:16 | HOLDOFF | 0xF | Sets the number of clock cycles for the memory controller to wait between any two non-burst accesses to Bank 2 |
| 15:12 | TIMER3 | 0xF | Sets the number of clock cycles between event 3 and event 4 for any Bank 2 access |
| 11:8 | TIMER2 | 0xF | Sets the number of clock cycles between event 2 and event 3 for any Bank 2 access |
| 7:4 | TIMER1 | 0xF | Sets the number of clock cycles between event 1 and event 2 for any Bank 2 access |
| 3:0 | TIMER0 | 0xF | Sets the number of clock cycles between event 0 and event 1 for any Bank 2 access |

### *7.2.11 Bank 2 Read Sequence (BANK2_READ)*

Offset: 0x0C004028
Reset Value: See field descriptions
Access: Read/Write

This register controls the sequence of events needed to execute a read of the device mapped to Bank 2. The default sequence is a standard SRAM-style access. CS is asserted on event 0; OE is asserted on event 1; OE is deasserted and data is captured on event 2; and CS is deasserted on event 3. In any sequence, the address is valid starting at event 0 and ending at either the END_EVENT or the BURST_END_EVENT.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31 | ENABLE_WAIT | 0x0 | If set, extends one of the four periods until the WAIT signal has been deasserted. The period to extend is selected by the WAIT_EVENT field in this register. |
| 30:28 | WAIT_EVENT | 0x2 | Indicates which event should delay based on the WAIT signal. Only used if the ENABLE_WAIT bit in this register is set. |
| 27 | RES | 0x0 | Reserved |
| 26:24 | END_EVENT | 0x3 | Indicates the final event in the read sequence. When this event is reached, all control signals return to their default states and the HOLDOFF counter starts. |
| 23 | RES | 0x0 | Reserved |
| 22:20 | BURST_END_EVENT | 0x7 | Indicates the final event in a burst read sequence. When this event is reached in the middle of a burst, the read sequence immediately jumps to the BURST_START_EVENT. If the value in this register is 0x7, bursting is disabled and consecutive burst requests complete as individual transactions (complete with holdoff). |
| 19 | RES | 0x0 | Reserved |
| 18:16 | BURST_START_EVENT | 0x7 | Indicates the start event of a burst read sequence. If the value in this register is 0x7, bursting is disabled and consecutive burst requests complete as individual transactions (complete with holdoff). |
| 15 | EVENT3_DC | 0x0 | If Set, the read data is captured on the same cycle as event 3 |
| 14 | EVENT3_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 3 and event 4 |
| 13 | EVENT3_OE | 0x0 | If set, the OE signal is set between event 3 and event 4. |
| 12 | EVENT3_CS | 0x0 | If set, the Bank 2 CS signal is set between event 3 and event 4. |
| 11 | EVENT2_DC | 0x1 | If Set, the read data is captured on the same cycle as event 2 |
| 10 | EVENT2_BE | 0x0 | If set, the BE signals (UBE/LBE) are set appropriately between event 2 and event 3 |
| 9 | EVENT2_OE | 0x0 | If set, the OE signal is set between event 2 and event 3. |
| 8 | EVENT2_CS | 0x1 | If set, the Bank 2 CS signal is set between event 2 and event 3 |
| 7 | EVENT1_DC | 0x0 | If set, the read data is captured on the same cycle as event 1 |
| 6 | EVENT1_BE | 0x0 | If set, the BE signals (UBE/LBE) are set appropriately between event 1 and event 2 |
| 5 | EVENT1_OE | 0x1 | If set, the OE signal is set between event 1 and event 2 |
| 4 | EVENT1_CS | 0x1 | If set, the Bank 2 CS signal is set between event 1 and event 2 |
| 3 | EVENT0_DC | 0x0 | If Set, the read data is captured on the same cycle as event 0 |
| 2 | EVENT0_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 0 and event 1 |
| 1 | EVENT0_OE | 0x0 | If set, the OE signal is set between event 0 and event 1 |
| 0 | EVENT0_CS | 0x1 | If set, the Bank 2 CS signal is set between event 0 and event 1 |

### *7.2.12 Bank 2 Write Sequence (BANK2_WRITE)*

Offset: 0x0C00402C
Reset Value: See field descriptions
Access: Read/Write

This register controls the sequence of events needed to execute a write of the device mapped to Bank 2. In any transaction, the address and write data are valid starting at event 0 and ending at either the END_EVENT or the BURST_END_EVENT.

| Bits | Bit Name | Reset Value | Description |
|---|---|---|---|
| 31 | ENABLE_WAIT | 0x0 | If set, extend one of the four periods until the WAIT signal deasserts. The WAIT_event field in this register selects the period to extend. |
| 30:28 | WAIT_EVENT | 0x2 | Indicates which event should be delayed based on the WAIT signal; only used if the ENABLE_WAIT bit in this register is set. |
| 27 | RES | 0x0 | Reserved |
| 26:24 | END_EVENT | 0x3 | Indicates the final event in the write sequence. When this event is reached, control signals return to their default states and the HOLDOFF counter starts. |
| 23 | RES | 0x0 | Reserved |
| 22:20 | BURST_END_EVENT | 0x7 | Indicates the final event in a burst write sequence. When this event is reached in the middle of a burst, the write sequence immediately jumps to the BURST_START_event. If the value in this register is 0x7, bursting is disabled and consecutive burst requests complete as individual transactions (complete with holdoff). |
| 19 | RES | 0x0 | Reserved |
| 18:16 | BURST_START_EVENT | 0x7 | Indicates the start event of a burst write sequence. If the value in this register is 0x7, bursting is disabled and consecutive burst requests complete as individual transactions (complete with holdoff). |
| 15 | RES | 0x0 | Reserved |
| 14 | EVENT3_BE | 0x0 | If set, the BE signals (UBE/LBE) are set appropriately between event 3 and event 4 |
| 13 | EVENT3_WE | 0x0 | If set, the WE signal is set between event 3 and event 4 |
| 12 | EVENT3_CS | 0x0 | If set, the Bank 2 CS signal is set between event 3 and event 4 |
| 11 | | 0x0 | Reserved |
| 10 | EVENT2_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 2 and event 3 |
| 9 | EVENT2_WE | 0x0 | If set, the WE signal is set between event 2 and event 3 |
| 8 | EVENT2_CS | 0x1 | If set, the Bank 2 CS signal is set between event 2 and event 3 |
| 7 | | 0x0 | Reserved |
| 6 | EVENT1_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 1 and event 2 |
| 5 | EVENT1_WE | 0x1 | If set, the WE signal is set between event 1 and event 2 |
| 4 | EVENT1_CS | 0x1 | If set, the Bank 2 CS signal is set between event 1 and event 2 |
| 3 | | 0x0 | Reserved |
| 2 | EVENT0_BE | 0x0 | If set, the BE signals (UBE/LBE) is set appropriately between event 0 and event 1 |
| 1 | EVENT0_WE | 0x0 | If set, the WE signal is set between event 0 and event 1 |
| 0 | EVENT0_CS | 0x1 | If set, the Bank 2 CS signal is set between event 0 and event 1 |

### 7.2.13 Interrupt When Timing Margin Small (TIMING_INT_ENABLE)

Offset: 0x0C004344
Reset Value: 0x0
Access: Read/Write

This register enables an interrupt when a particular bit in the timing vector is set. Software can program this register to cause an interrupt if the timing margin becomes too low. The vector is a number with leading zeroes.

- 00000001 = Most conservative setting (maximum margin)

...

- 01111111 = Most aggressive setting (minimal margin)

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7:0 | VECTOR | Timing vector interrupt enable |

### 7.2.14 MC Interrupt Bits Status (MC_ERROR_STATUS)

Offset: 0x0C004348
Reset Value: 0x0
Access: Read only

This register drives the MC error interrupt signal to the CPU.

| Bits | Bit Name | Description |
|---|---|---|
| 31:2 | RES | Reserved |
| 1 | AHB | AHB error bit is set, AHB error occurred. See the MC AHB error registers for details. |
| 0 | TIMING | Timing error interrupt occurred. |

## 7.3 UART Registers

Table 7-3 summarizes UART registers.

*Table 7-3.* **UART Registers**

| Offset | Name | Description | Page |
|---|---|---|---|
| 0x0C008000 | RBR | Receive buffer | page 88 |
| 0x0C008000 | THR | Transmit holding | page 88 |
| 0x0C008000 | DLL | Divisor latch low | page 88 |
| 0x0C008004 | DLH | Divisor latch high | page 88 |
| 0x0C008004 | IER | Interrupt enable | page 89 |
| 0x0C008008 | IIR | Interrupt identity | page 89 |
| 0x0C008008 | FCR | FIFO control | page 89 |
| 0x0C00800C | LCR | Line control | page 90 |
| 0x0C008010 | MCR | Modem control | page 90 |
| 0x0C008014 | LSR | Line status | page 91 |
| 0x0C008018 | MSR | Modem status | page 92 |

### 7.3.1 Receive Buffer (RBR)

Offset:0x0C008000
Reset Value: 0x0
Access: Read only

This register contains the data byte received on the serial input port (sin). The data in this register is valid only if the data ready (DR) bit in the line status register (LSR) is set.

In non-FIFO mode (FIFO_mode = 0), the data must be read before the next data arrives, otherwise it will be overwritten, resulting in an overrun error.

In FIFO mode (FIFO_mode = 1), this register accesses the head of the Rx FIFO. If the Rx FIFO is full and the register is not read before the next data character arrives, the data already in the FIFO is preserved but any incoming data is lost and an overrun error occurs.

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7:0 | RBR | Receive buffer value |

### 7.3.2 Transmit Holding (THR)

Offset:0x0C008000
Reset Value: 0x0
Access: Write only

This register contains data to transmit on the serial output port (sout). Data can be written to this register fi the THR empty (THRE) bit of the line status register (LSR) is set. If FIFOs are not enabled and THRE is set, writing a single character to the THR clears the THRE.

Any additional writes to the THR before the THRE is set again overwrites THR data. If FIFOs are enabled and THRE is set, up to 16 characters of data may be written to the THR before the FIFO is full.

Any attempt to write data when the FIFO is full results in lost write data.

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7:0 | THR | Transmit buffer value |

### 7.3.3 Divisor Latch Low (DLL)

Offset:0x0C008000
Reset Value: 0x0
Access: Read/Write

This register, together with the "Divisor Latch High (DLH)" register, forms a 16-bit divisor latch register containing the UART baud rate divisor accessed by first setting DLAB bit [7] in the line control register (LCR).

The output baud rate is equal to the input clock frequency divided by sixteen times the value of the baud rate divisor:

baud = (clock freq) / (16 * divisor)

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7:0 | DLL | Divisor latch low |

### 7.3.4 Divisor Latch High (DLH)

Offset:0x0C008004
Reset Value: 0x0
Access: Read/Write

This register together with the "Divisor Latch Low (DLL)" register forms a 16-bit divisor latch register containing the UART baud rate divisor, accessed by first setting DLAB bit [7] in the line control register (LCR).

The output baud rate is equal to the input clock frequency divided by sixteen times the value of the baud rate divisor:

baud = (clock freq) / (16 * divisor)

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7:0 | DLH | Divisor latch high |

### 7.3.5 Interrupt Enable (IER)

Offset:0x0C008004
Reset Value: 0x0
Access: Read/Write

This register contains the four bits that enable interrupt generation.

| Bits | Bit Name | Description |
|---|---|---|
| 31:4 | RES | Reserved |
| 3 | EDDSI | Enable modem status interrupt |
| 2 | ELSI | Enable receiver line status interrupt |
| 1 | ETBEI | Enable register empty interrupt |
| 0 | ERBFI | Enable received data available interrupt |

### 7.3.6 Interrupt Identity (IIR)

Offset:0x0C008008
Reset Value: 0x0
Access: Read only

This register identifies the source of an interrupt. The upper two bits of the register are FIFO-enabled.

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7:6 | FIFO_STATUS | FIFO enable status bits.<br>■ 00 = FIFO disabled<br>■ 11 = FIFO enabled |
| 5:4 | RES | Reserved |
| 3:0 | IID | Identifies interrupt source.<br>■ 0000 = Modem Status Changed<br>■ 0001 = No interrupt pending<br>■ 0010 = THR empty<br>■ 0100 = Received data available<br>■ 0110 = Receiver status<br>■ 1100 = Character time out |

### 7.3.7 FIFO Control (FCR)

Offset:0x0C008008
Reset Value: 0x0
Access: Write only

If FIFO_mode = 0, this register has no effect. If FIFO_mode = 1, this register controls read and write data FIFO operation and operation mode for the Txrdy_n and Rxrdy_n DMA signals. Bit [0] enables Tx and Rx FIFOs. Writing a 1 to bit [1] resets and flushes data in the Rx FIFO.

Writing a 1 to bit [2] resets and flushes data in the Tx FIFO. The FIFOs are also reset if bit [0] changes value. If FIFO_mode = 1 and bit [0] is set to 1, bits [3, 6, 7] are active.

Bit [3] determines the DMA signalling mode for Txrdy_n and Rxrdy_n output signals.

Bit [6] and bit [7] set the trigger level in the Rx FIFO for both the Rxrdy_n signal and the enable received data available interrupt. Return the current time values.

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7:6 | RCVR_TRIG | Sets the trigger level in the Rx FIFO for both the Rxrdy_n signal and the enable received data available interrupt.<br>■ 00 = 1 byte in FIFO<br>■ 01 = 4 bytes in FIFO<br>■ 10 = 8 bytes in FIFO<br>■ 11 = 14 bytes in FIFO |
| 5:4 | RES | Reserved |
| 3 | DMA_MODE | Determines DMA signalling mode for the Txrdy_n and Rxrdy_n output signals |
| 2 | XMIT_FIFO_RST | Writing this bit resets and flushes data in the Tx FIFO |
| 1 | RCVR_FIFO_RST | Writing this bit resets and flushes data in the Rx FIFO |
| 0 | FIFO_EN | Enables the Tx and Rx FIFOs. FIFOs also reset anytime this bit changes value. |

### 7.3.8 Line Control (LCR)

This register controls the format of data transmitted and received by the UART controller.

Offset:0x0C00800C
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7 | DLAB | Divisor latch address<br>Setting this bit enables reading and writing of the Divisor latch register (the DLL with the DLH) to set the UART baud rate. This bit must be cleared after initial baud rate setup to access other registers. |
| 6 | BREAK | Setting this bit sends a break signal by holding the sout line low (when not in loopback mode, as determined by modem control register bit [4]), until the bit clears. In loopback mode, the break condition loops internally back to the receiver. |
| 5 | RES | Reserved |
| 4 | EPS | Even parity select<br>If parity is enabled, this bit selects between even and odd parity. If this bit is a logic 1, an even number of logic 1s is transmitted or checked. If this bit is a logic 0, an odd number of logic 1s is transmitted or checked. |
| 3 | PEN | Parity is enabled When set |
| 2 | STOP | Controls the number of stop bits transmitted. If bit 2 is a logic 0, one stop bit is transmitted in serial data. If bit 2 is a logic 1 and the data bits are set to 5, one and a half stop bits are generated. Otherwise, two stop bits are generated and transmitted in the serial data out. |
| 1:0 | CLS | Controls the number of bits per character.<br>■ 00 = 5 bits<br>■ 01 = 6 bits<br>■ 10 = 7 bits<br>■ 11 = 8 bits |

### 7.3.9 Modem Control (MCR)

This register controls the modem interface.

Offset:0x0C008010
Reset Value: 0x1
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:6 | RES | Reserved |
| 5 | LOOPBACK | When set, data on the sout line is held HIGH, while serial data output loops back internally to the sin line. In this mode all interrupts are fully functional. Used for diagnostic purposes. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out1_n) loop back internally to the inputs. |
| 4 | RES | Reserved |
| 3 | OUT2 | Drives UART output UART_OUT2_L |
| 2 | OUT1 | Drives UART output UART_OUT1_L |
| 1 | RTS | Drives UART output RTS_L |
| 0 | DTR | Drives UART output DTR_L |

### *7.3.10 Line Status (LSR)*

Offset:0x0C008014
Reset Value: 0x0
Access: Read/Write

This register contains receiver and transmitter data transfer status. This status can be read by the programmer at any time.

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7 | FERR | Error in the Rx FIFO (FERR)<br>Only active when FIFOs are enabled. It is set when at least one parity error, framing error, or break indication exists in the FIFO. This bit clears when the LSR is read AND the character with the error is at the top of the Rx FIFO AND no subsequent errors exist in the FIFO. |
| 6 | TEMT | Transmitter empty (TEMT)<br>In FIFO mode, this bit is set when the transmitter shift register and the FIFO are both empty. In non-FIFO mode, this bit is set when the transmitter holding register and the transmitter shift register are both empty. |
| 5 | THRE | Transmitter Holding Register Empty (THRE)<br>When set, indicates the UART controller can accept new characters for transmission. This bit is set when data transfers from THR to the transmitter shift register and no new data is written to THR. The bit also causes a THRE interrupt, if the THRE interrupt is enabled. |
| 4 | BI | Break Interrupt (BI)<br>Set when the serial input (sin) is held in a logic zero state for longer than the sum of start time + data bits + parity + stop bits. A break condition on sin causes the UART to receive one and only one character consisting of all zeros. In FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears this bit. In the non FIFO mode, the BI indication occurs immediately and persists until the LSR is read. |
| 3 | FE | Framing Error (FE)<br>Set when a framing error exists in the receiver. A framing error occurs when the receiver detect no valid STOP bits in received data. In FIFO mode, because the framing error is associated to a received character, it is revealed when the character with the framing error is at the top of the FIFO. The OE, PE, and FE bits reset when a read of the LSR is performed. |
| 2 | PE | Parity Error (PE)<br>Set when a parity error exists in the receiver if the LCR parity enable (PEN) is set. In FIFO mode, because the parity error is associated to a received character, it is revealed when the character with the parity error arrives at the top of the FIFO. |
| 1 | OE | Overrun error (OE)<br>When set, indicates an overrun error has occurred because a new data character was received before previous data was read. In non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR, overwriting data in the RBR. In FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. FIFO data is retained and data in the receive shift register is lost. |
| 0 | DR | Data Ready (DR)<br>When set, indicates the receiver contains at least one character in the RBR or the Rx FIFO. This bit is cleared when the RBR is read in non-FIFO mode, or when the Rx FIFO is empty, in FIFO mode. |

### *7.3.11 Modem Status (MSR)*

This register contains the current status of the modem control input lines and if they changed.

Offset:0x0C008018
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7 | DCD | Contains information on the current state of the modem control lines. DCD (bit [7]) is the compliment of DCD_L |
| 6 | RI | Contains information on the current state of the modem control lines. RI (bit [6]) is the compliment of RI_L |
| 5 | DSR | Contains information on the current state of the modem control lines. DSR (bit [5]) is the compliment of DSR_L |
| 4 | CTS | Contains information on the current state of the modem control lines. CTS (bit 4]) is the compliment of CTS_L |
| 3 | DDCD | Records whether the modem control line DCD_L has changed since the last time the CPU read the MSR |
| 2 | TERI | Indicates RI_L has changed from an active low, to an inactive high state since the last time the MSR was read |
| 1 | DDSR | Records whether the modem control line DSR_L has changed since the last time the CPU read the MSR |
| 0 | DCTS | Records whether the modem control line CTS_L has changed since the last time the CPU read the MSR |

## *7.4 Serial Interface Registers*

Table 7-4 summarizes serial interface registers.

*Table 7-4.* **Serial Interface Registers**

| Address | Name | Description | Page |
|---|---|---|---|
| 0x0C00C000 | SI_CONFIG | Serial interface configuration | page 93 |
| 0x0C00C004 | SI_CS | Serial interface control/status | page 94 |
| 0x0C00C008 | SI_TXDATA0 | First four bytes of Tx data | page 94 |
| 0x0C00C00C | SI_TXDATA1 | Second four bytes of Tx data | page 95 |
| 0x0C00C010 | SI_RXDATA0 | First four bytes of Rx data | page 95 |
| 0x0C00C014 | SI_RXDATA1 | Second four bytes of Rx data | page 95 |

### 7.4.1 SI Configuration (SI_CONFIG)

Offset:0x0C00C000
Reset Value: 0x0
Access: Read/Write

When this register is written to, the serial interface (SI) clock has 1/divider of a period on the bus clock. This register also controls which clock edge data is driven and sampled on, as well as the serial interface properties, such as whether the data port has an external pullup and thus behaves like a pseudo-open-drain, and whether the interface is $I^2C$ or SPI.

| Bits | Bit Name | Description |
|---|---|---|
| 31:20 | RES | Reserved |
| 19 | ERR_INT | Determines whether DONE_ERR triggers an interrupt |
| 18 | BIDIR_OD_DATA | Determines whether bidirectional data pin si_si is pseudo-open-drain. If it is, then it is only driven low when data is low and not driven when data is high. This is only applicable for $I^2C$ interface where the data pin is bidirectional This bit has no effect for SPI interface.<br>■ 0 = Data out pin is not pseudo-open-drain<br>■ 1 = Data out pin is pseudo-open-drain |
| 17 | RES | Reserved |
| 16 | I2C | determines whether the serial interface is a $I^2C$ or SPI interface<br>■ 0 = SI is SPI<br>■ 1 = SI is $I^2C$ |
| 15:8 | | Reserved |
| 7 | POS_SAMPLE | Determines whether data are sampled on the positive edge of SI clock<br>■ 0 = Sample on negative edge<br>■ 1 = Sample on positive edge |
| 6 | POS_DRIVE | Determines whether data are driven on the negative edge of SI clock<br>■ 0 = Drive on negative edge<br>■ 1 = Drive on positive edge |
| 5 | INACTIVE_DATA | Determines the value of inactive data out<br>■ 0 = Inactive data is deasserted<br>■ 1 = Inactive data is asserted |
| 4 | INACTIVE_CLK | Determines the value of inactive clock<br>■ 0 = Inactive clock is deasserted<br>■ 1 = Inactive clock is asserted |
| 3 | | Reserved |
| 2:0 | DIVIDER | Determine the value of the clock divider for si_ck. The core clock to be divided is 38.4 MHz.<br>■ 0 = Divide by 2<br>■ 1 = Divide by 4<br>■ 2 = Divide by 8<br>■ 3 = Divide by 16<br>■ 4 = Divide by 32<br>■ 5 = Divide by 64 |

### 7.4.2 SI Control/Status (SI_CS)

Offset:0x0C00C004
Reset Value: 0x0
Access: Read/Write

This register is used to control the serial interface and to keep status of the serial interface.

| Bits | Bit Name | Description |
|---|---|---|
| 31:14 | RES | Reserved |
| 13:11 | BIT_CNT_IN_LAST_BYTE | Determine the number of bits (from 1 to 8) to send or to receive on the serial interface for the last byte.<br>■ 1 = 1 bit<br>■ 2 = 2 bits<br>...<br>■ 7 = 7 bits<br>■ 0 = 8 bits |
| 10 | DONE_ERR | Set by the serial interface logic when the current sequence completes with error. Cleared when the START bit is set in the SI_CS register and set when the entire sequence (Tx and Rx) has completed. |
| 9 | DONE_INT | Set by the SI logic when the current sequence completes. Cleared when 1 is written to this field or when the START bit is set in the SI_CS register and set when the entire sequence (Tx and Rx) has completed. Setting this bit triggers an interrupt if the interrupt enable bit for the serial interface in the interrupt control register is asserted. When this bit is polled, clear the interrupt enable so no interrupt is generated. |
| 8 | START | Setting this bit starts a Tx/Rx sequence on the serial interface. TXCNT bytes are transmitted on the interface and then RXCNT bytes are received. This bit clears right after it is set. |
| 7:4 | RXCNT | Determine the number of bytes (from 0 to 8) to receive on the serial interface. Receive is started when the START bit is set. The chip select stays asserted and the clock continues running to receive from 0 to 8 bytes of data. The first byte loads into bits [7:0] of the SI_RXDATA0 register and the eighth byte (if needed) into bits [31:24] of the SI_RXDATA1 register. No data transmits during the receive phase. |
| 3:0 | TXCNT | Determine the number of bytes (from 0 to 8) to send on the serial interface. Data, starting with the DATA0 field of the SI_TXDATA0 register and ending with the DATA7 field of the SI_TXDATA1 register, is sent out when the START bit is set. The chip select signal asserts and the clock runs for the entire transmit. No data is received during the transmit. |



### 7.4.3 First Four Bytes of Tx Data (SI_TXDATA0)

Offset:0x0C00C008
Reset Value: 0x0
Access: Read/Write

This register, combined with the "Second Four Bytes of Tx Data (SI_TXDATA1)" register, contains the data bits to send out on the serial interface. Data is sent, starting with bits [7:0] of this register and ending with bits [31:24] of SI_TXDATA1, when the START bit is set in the SI_CS register. The bits in each byte are sent out serially with the most significant bit sent first.

| Bits | Bit Name | Description |
|---|---|---|
| 31:24 | DATA3 | Fourth byte transferred<br>Sent if the TXCNT field of the SI_CS register is greater than 3. |
| 23:16 | DATA2 | Third byte transferred<br>Sent if the TXCNT field of the SI_CS register is greater than 2. |
| 15:8 | DATA1 | Second byte transferred<br>Sent if the TXCNT field of the SI_CS register is greater than 1. |
| 7:0 | DATA0 | First byte transferred<br>Sent if the TXCNT field of the SI_CS register is greater than 0. |

### 7.4.4 Second Four Bytes of Tx Data (SI_TXDATA1)

Offset:0x0C00C00C
Reset Value: 0x0
Access: Read/Write

This register, combined with the "First Four Bytes of Tx Data (SI_TXDATA0)" register, contains the data bits to send out on the serial interface. Data is sent, starting with bits [7:0] of this register and ending with bits [31:24] of SI_TXDATA0, when the START bit is set in the SI_CS register. The bits in each byte are sent out serially with the most significant bit sent first.

| Bits | Bit Name | Description |
|---|---|---|
| 31:24 | DATA7 | Eighth byte transferred<br>Sent if the TXCNT field of the SI_CS register is greater than 7. |
| 23:16 | DATA6 | Seventh byte transferred<br>Sent if the TXCNT field of the SI_CS register is greater than 6. |
| 15:8 | DATA5 | Sixth byte transferred<br>Sent if the TXCNT field of the SI_CS register is greater than 5. |
| 7:0 | DATA4 | Fifth byte transferred<br>Sent if the TXCNT field of the SI_CS register is greater than 4. |

### 7.4.5 First Four Bytes of Rx Data (SI_RXDATA0)

Offset:0x0C00C010
Reset Value: 0x0
Access: Read only

This register, combined with "Second Four Bytes of Rx Data (SI_RXDATA1)", captures data bits from the serial interface after the transmit completes. The first byte of data is placed in bits [7:0] of this register, and the last byte of data in bits [31:24] of SI_RXDATA1. The bits in each byte are captured serially with the most significant bit captured first.

| Bits | Bit Name | Description |
|---|---|---|
| 31:24 | DATA3 | Fourth byte received<br>Loaded if the RXCNT field of the SI_CS register is greater than 3. |
| 23:16 | DATA2 | Third byte received<br>Loaded if the RXCNT field of the SI_CS register is greater than 2. |
| 15:8 | DATA1 | Second byte received<br>Loaded if the RXCNT field of the SI_CS register is greater than 1. |
| 7:0 | DATA0 | First byte received<br>Loaded if the RXCNT field of the SI_CS register is greater than 0. |

### 7.4.6 Second Four Bytes of Rx Data (SI_RXDATA1)

Offset:0x0C00C014
Reset Value: 0x0
Access: Read only

This register, combined with "First Four Bytes of Rx Data (SI_RXDATA0)", captures data bits from the serial interface after the transmit completes. The first byte of data is placed in bits [7:0] of this register, and the last byte of data in bits [31:24] of SI_RXDATA0. The bits in each byte are captured serially with the most significant bit captured first.

| Bits | Bit Name | Description |
|---|---|---|
| 31:24 | DATA7 | Eighth byte received<br>Loaded if the RXCNT field of the SI_CS register is greater than 7. |
| 23:16 | DATA6 | Seventh byte received<br>Loaded if the RXCNT field of the SI_CS register is greater than 6. |

| Bits | Bit Name | Description |
|---|---|---|
| 15:8 | DATA5 | Sixth byte received<br>Loaded if the RXCNT field of the SI_CS register is greater than 5. |
| 7:0 | DATA4 | Fifth byte received<br>Loaded if the RXCNT field of the SI_CS register is greater than 4. |

## 7.5 GPIO Registers

Table 7-5 summarizes GPIO registers.

*Table 7-5.* **GPIO Registers**

| Offset | Name | Description | Page |
|---|---|---|---|
| 0x0C010000 | GPIO_OUT | Drive data out on GPIO pins | page 96 |
| 0x0C010004 | GPIO_OUT_W1TS | Write a 1 to set the GPIO_OUT alias | page 98 |
| 0x0C010008 | GPIO_OUT_W1TC | Write a 1 to clear the GPIO_OUT alias | page 98 |
| 0x0C01000C | GPIO_ENABLE | Enable output drivers for GPIO pins | page 98 |
| 0x0C010010 | GPIO_ENABLE_W1TS | Write a 1 to set the GPIO_ENBLE alias | page 98 |
| 0x0C010014 | GPIO_ENABLE_W1TC | Write a 1 to clear the GPIO_ENABLE alias | page 99 |
| 0x0C010018 | GPIO_IN | Sample data on GPIO pins | page 99 |
| 0x0C01001C | GPIO_STATUS | GPIO pins interrupt status | page 99 |
| 0x0C010020 | GPIO_STATUS_W1TS | Write a 1 to set the GPIO_STATUS alias | page 99 |
| 0x0C010024 | GPIO_STATUS_W1TC | Write a 1 to clear the GPIO_STATUS alias | page 100 |
| 0x0C010028 | GPIO_PIN0 | GPIO 0 configuration | page 100 |
| 0x0C01002C | GPIO_PIN1 | GPIO 1 configuration | page 101 |
| 0x0C010030 | GPIO_PIN2 | GPIO 2 configuration | page 102 |
| 0x0C010034 | GPIO_PIN3 | GPIO 3 configuration | page 103 |
| 0x0C010038 | GPIO_PIN4 | GPIO 4 configuration | page 104 |
| 0x0C01003C | GPIO_PIN5 | GPIO 5 configuration | page 105 |
| 0x0C010040 | GPIO_PIN6 | GPIO 6 configuration | page 106 |
| 0x0C010044 | GPIO_PIN7 | GPIO 7 configuration | page 107 |
| 0x0C010048 | GPIO_PIN8 | GPIO 8 configuration | page 108 |
| 0x0C01004C | GPIO_PIN9 | GPIO 9 configuration | page 109 |
| 0x0C010050 | GPIO_PIN10 | GPIO 10 configuration | page 110 |
| 0x0C010054 | GPIO_PIN11 | GPIO 11 configuration | page 111 |
| 0x0C010058 | GPIO_PIN12 | GPIO 12 configuration | page 112 |
| 0x0C01005C | GPIO_PIN13 | GPIO 13 configuration | page 113 |
| 0x0C010060 | GPIO_PIN14 | GPIO 14 configuration | page 114 |
| 0x0C010064 | GPIO_PIN15 | GPIO 15 configuration | page 115 |
| 0x0C010068 | GPIO_PIN16 | GPIO 16 configuration | page 116 |
| 0x0C01006C | GPIO_PIN17 | GPIO 17 configuration | page 117 |
| 0x0C010070 | SDIO_PIN | SDIO pin driver configuration | page 118 |
| 0x0C010074 | CLK_REQ_PIN | CLK_REQ pin driver configuration | page 118 |
| 0x0C010078 | SIGMA_DELTA | Sigma delta PWM configuration | page 119 |

### 7.5.1 Drive Data Out on GPIO Pins (GPIO_OUT)

Offset:0x0C010000
Reset Value: 0x0
Access: Read/Write

When software writes this register, the write data is driven out for those pins enabled by the GPIO_ENABLE register.

| Bits | Bit Name | Description |
|---|---|---|
| 31:18 | RES | Reserved |
| 17:0 | DATA | GPIO Pin data output<br>■ 0 = Drive the pin low<br>■ 1 = Drive the pin high |

### 7.5.2 Write 1 to Set GPIO_OUT Alias (GPIO_OUT_W1TS)

Offset:0x0C010004
Reset Value: 0x0
Access: Write only

This register is an alias of GPIO_OUT, which allows software to set selected bits without changing the value of other bits. For example, a write of 0x5 to this register sets bits 0 and 2 in GPIO_OUT, but all other bits in GPIO_OUT will remain unchanged.

| Bits | Bit Name | Description |
|---|---|---|
| 31:18 | RES | Reserved |
| 17:0 | DATA | For each bit position in the data word:<br>■ 0 = Do not change the bit<br>■ 1 = Set the bit to 1 |

### 7.5.3 Write 1 to Clear GPIO_OUT Alias (GPIO_OUT_W1TC)

Offset:0x0C010008
Reset Value: 0x0
Access: Write only

This register is an alias of GPIO_OUT, which allows software to clear selected bits without changing the value of other bits. For example, a write of 0x5 to this register clears bits 0 and 2 in GPIO_OUT, but all other bits in GPIO_OUT remain unchanged.

| Bits | Bit Name | Description |
|---|---|---|
| 31:18 | RES | Reserved |
| 17:0 | DATA | For each bit position in the data word:<br>■ 0 = Do not change the bit<br>■ 1 = Clear the bit to 0 |

### 7.5.4 Enable Output Drivers for GPIO Pins (GPIO_ENABLE)

Offset:0x0C01000C
Reset Value: 0x0
Access: Read/Write

When software writes this register, bits set to 1 enable the GPIO output driver for the corresponding GPIO pin. When the output driver is enabled, the corresponding GPIO_DATA_OUT bit or selected sigma delta pulse waveform modulator (PWM) are driven to the pin. When the output driver is not enabled, no value is driven to the pin.

| Bits | Bit Name | Description |
|---|---|---|
| 31:18 | RES | Reserved |
| 17:0 | DATA | GPIO Pin data output enable.<br>■ 0 = Driver not enabled for the corresponding pin<br>■ 1 = Driver enabled for the corresponding pin |

### 7.5.5 Write 1 to Set GPIO_ENABLE Alias (GPIO_ENABLE_W1TS)

Offset:0x0C010010
Reset Value: 0x0
Access: Write only

This register is an alias of GPIO_ENABLE, which allows software to set selected bits without changing the value of other bits. For example, a write of 0x5 to this register sets bits 0 and 2 in GPIO_ENABLE, but all other bits in GPIO_ENABLE remain unchanged.

| Bits | Bit Name | Description |
|---|---|---|
| 31:18 | RES | Reserved |
| 17:0 | DATA | For each bit position in the data word:<br>■ 0 = Do not change the bit<br>■ 1 = Set the bit to 1 |

### 7.5.6 Write 1 to Clear GPIO_ENABLE Alias (GPIO_ENABLE_W1TC)

Offset:0x0C010014
Reset Value: 0x0
Access: Write only

This register is an alias of GPIO_ENABLE, which allows software to clear selected bits without changing the value of other bits. For example, a write of 0x5 to this register clears bits 0 and 2 in GPIO_ENABLE, but all other bits in GPIO_ENABLE remain unchanged.

| Bits | Bit Name | Description |
|---|---|---|
| 31:18 | RES | Reserved |
| 17:0 | DATA | For each bit position in the data word:<br>■ 0 = Do not change the bit<br>■ 1 = Clear the bit to 0 |

### 7.5.7 Sample Data on GPIO Pins (GPIO_IN)

Offset:0x0C010018
Reset Value: 0x0
Access: Read only

A software read of this register returns the current data values at the GPIO pin input registers.

| Bits | Bit Name | Description |
|---|---|---|
| 31:18 | RES | Reserved |
| 17:0 | DATA | GPIO Pin data input |

### 7.5.8 GPIO Pins Interrupt Status (GPIO_STATUS)

Offset:0x0C01001C
Reset Value: 0x0
Access: Read/Write

GPIO pin transitions can cause interrupts to be set in this register. The transition type that causes interrupt is set in GPIO_PIN.

| Bits | Bit Name | Description |
|---|---|---|
| 31:18 | RES | Reserved |
| 17:0 | INTERRUPT | GPIO Pin Interrupt pending.<br>■ 0 = Interrupt not pending.<br>■ 1 = Interrupt pending. Remains set until software explicitly clears this bit. |

### 7.5.9 Write 1 to Set GPIO_STATUS Alias (GPIO_STATUS_W1TS)

Offset:0x0C010020
Reset Value: 0x0
Access: Write only

This register is an alias of GPIO_STATUS, which allows software to set selected bits without changing the value of other bits. For example, a write of 0x5 to this register sets bits 0 and 2 in GPIO_STATUS, but all other bits in GPIO_STATUS remain unchanged.

| Bits | Bit Name | Description |
|---|---|---|
| 31:18 | RES | Reserved |
| 17:0 | INTERRUPT | For each bit position in the data word:<br>■ 0 = Do not change the bit<br>■ 1 = Set the bit to 1 |

### 7.5.10 Write 1 to Clear GPIO_STATUS Alias (GPIO_STATUS_W1TC)

Offset:0x0C010024
Reset Value: 0x0
Access: Write only

This register is an alias of GPIO_STATUS which allows software to clear selected bits without changing the value of other bits. For example, a write of 0x5 to this register will clear bits 0 and 2 in GPIO_STATUS, but all other bits in GPIO_STATUS will remain unchanged.

| Bits | Bit Name | Description |
|---|---|---|
| 31:18 | RES | Reserved |
| 17:0 | INTERRUPT | For each bit position in the data word:<br>1 = clear the bit to 0.<br>0 = do not change the bit. |

### 7.5.11 GPIO 0 Configuration (GPIO_PIN0)

Configures the pin type and interrupt behavior.

Offset:0x0C010028
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = $I^2C$/SPI<br>■ 1 = GPIO<br>■ 2 = Reserved<br>■ 3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin will also be sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:7 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 0:<br>■ 0 = Interrupt disabled for GPIO 0<br>■ 1 = Interrupt on 0 → 1 edge of GPIO pin 0<br>■ 2 = Interrupt on 1 → 0 edge of GPIO pin 0<br>■ 3 = Interrupt on any edge of GPIO pin 0<br>■ 4 = Interrupt on level 0 of GPIO pin 0<br>■ 5 = Interrupt on level 1 of GPIO pin 0<br>■ 6:7 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 0 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 0<br>■ 0 = GPIO_OUT register.<br>■ 1 = Sigma delta pulse waveform modulator (PWM) resource 0 |

### *7.5.12 GPIO 1 Configuration (GPIO_PIN1)*

Configures the pin type and interrupt behavior.

Offset:0x0C01002C
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = $I^2C$/SPI<br>■ 1 = GPIO<br>■ 2 = Reserved<br>■ 3 = Reserved |
| 10 | WAKEUP_ ENABLE | When set, an interrupt caused by this GPIO pin will also be sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:7 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 1:<br>■ 0 = Interrupt disabled for GPIO 1<br>■ 1 = Interrupt on 0 → 1 edge of GPIO pin 1<br>■ 2 = Interrupt on 1 → 0 edge of GPIO pin 1<br>■ 3 = Interrupt on any edge of GPIO pin 1<br>■ 4 = Interrupt on level 0 of GPIO pin 1<br>■ 5 = Interrupt on level 1 of GPIO pin 1<br>■ 6:7 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 1 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 1<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.13 GPIO 2 Configuration (GPIO_PIN2)*

Configures the pin type and interrupt behavior.

Offset:0x0C010030
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = UART<br>■ 1 = SPI<br>■ 2 = GPIO<br>■ 3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin will also be sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup.<br>■ 1 = Interrupt on this pin causes SLEEP wakeup. |
| 9:7 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 2:<br>■ 0 = Interrupt disabled for GPIO 2.<br>■ 1 = Interrupt on 0->1 edge of GPIO pin 2<br>■ 2 = Interrupt on 1->0 edge of GPIO pin 2<br>■ 3 = Interrupt on any edge of GPIO pin 2<br>■ 4 = Interrupt on level 0 of GPIO pin 2<br>■ 5 = Interrupt on level 1 of GPIO pin 2<br>■ 6:7 = Reserved |
| 6:3 | | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 2 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | | Reserved |
| 0 | SOURCE | Output source for GPIO pin 2<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.14 GPIO 3 Configuration (GPIO_PIN3)*

Configures pin type and interrupt behavior.

Offset:0x0C010034
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = UART<br>■ 1 = SPI<br>■ 2 = GPIO<br>■ 3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin will also be sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup.<br>■ 1 = Interrupt on this pin causes SLEEP wakeup. |
| 9:7 | INT_TYPE | An interrupt is set if the following occurs on GPIO pin 3:<br>■ 0 = Interrupt disabled for GPIO 3<br>■ 1 = Interrupt on 0 → 1 edge of GPIO pin 3<br>■ 2 = Interrupt on 1 → 0 edge of GPIO pin 3<br>■ 3 = Interrupt on any edge of GPIO pin 3<br>■ 4 = Interrupt on level 0 of GPIO pin 3<br>■ 5 = Interrupt on level 1 of GPIO pin 3<br>■ 6:7 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 3 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 3.<br>■ 0 = GPIO_OUT register.<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.15 GPIO 4 Configuration (GPIO_PIN4)*

Configures the pin type and interrupt behavior.

Offset:0x0C010038
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = Bluetooth interface<br>■ 1 = GPIO<br>■ 2:3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin will also be sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup.<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:7 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 4:<br>■ 0 = Interrupt disabled for GPIO 4<br>■ 1 = Interrupt on 0 → edge of GPIO pin 4<br>■ 2 = Interrupt on 1 → 0 edge of GPIO pin 4<br>■ 3 = Interrupt on any edge of GPIO pin 4<br>■ 4 = Interrupt on level 0 of GPIO pin 4<br>■ 5 = Interrupt on level 1 of GPIO pin 4<br>■ 6:7 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 4 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 4<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.16 GPIO 5 Configuration (GPIO_PIN5)*

Configures the pin type and interrupt behavior.

Offset:0x0C01003C
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = Bluetooth interface<br>■ 1 = GPIO<br>■ 2:3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:7 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 5:<br>■ 0 = Interrupt disabled for GPIO 5<br>■ 1 = Interrupt on 0 → 1 edge of GPIO pin 5<br>■ 2 = Interrupt on 1 → 0 edge of GPIO pin 5<br>■ 3 = Interrupt on any edge of GPIO pin 5<br>■ 4 = Interrupt on level 0 of GPIO pin 5<br>■ 5 = Interrupt on level 1 of GPIO pin 5<br>■ 6:7 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 5 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 5<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.17 GPIO 6 Configuration (GPIO_PIN6)*

Configures the pin type and interrupt behavior.

Offset:0x0C010040
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = Bluetooth interface<br>■ 1 = GPIO<br>■ 2:3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup.<br>■ 1 = Interrupt on this pin causes SLEEP wakeup. |
| 9:7 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 6:<br>■ 0 = Interrupt disabled for GPIO 6.<br>■ 1 = Interrupt on 0 → 1 edge of GPIO pin 6<br>■ 2 = Interrupt on 1 → 0 edge of GPIO pin 6<br>■ 3 = Interrupt on any edge of GPIO pin 6<br>■ 4 = Interrupt on level 0 of GPIO pin 6<br>■ 5 = Interrupt on level 1 of GPIO pin 6<br>■ 6:7 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO pin 6 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | GPIO pin 6 output source.<br>■ 0 = GPIO_OUT register.<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.18 GPIO 7 Configuration (GPIO_PIN7)*

Configures the pin type and interrupt behavior.

Offset:0x0C010044
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = Bluetooth interface<br>■ 1 = GPIO<br>■ 2:3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:7 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 7:<br>■ 0 = Interrupt disabled for GPIO 7.<br>■ 1 = Interrupt on 0 → edge of GPIO pin 7<br>■ 2 = Interrupt on 1 → edge of GPIO pin 7<br>■ 3 = Interrupt on any edge of GPIO pin 7<br>■ 4 = Interrupt on level 0 of GPIO pin 7<br>■ 5 = Interrupt on level 1 of GPIO pin 7<br>■ 6:7 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 7 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 7<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.19 GPIO 8 Configuration (GPIO_PIN8)*

Configures the pin type and interrupt behavior.

Offset:0x0C010048
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = GPIO<br>■ 1 = I2S_MCK (digital audio master clock out)<br>■ 2:3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:8 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 8:<br>■ 0 = Interrupt disabled for GPIO 8.<br>■ 1 = Interrupt on 0 → edge of GPIO pin 8<br>■ 2 = Interrupt on 1 → edge of GPIO pin 8<br>■ 3 = Interrupt on any edge of GPIO pin 8<br>■ 4 = Interrupt on level 0 of GPIO pin 8<br>■ 5 = Interrupt on level 1 of GPIO pin 8<br>■ 6:8 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 8 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 8<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.20 GPIO 9 Configuration (GPIO_PIN9)*

Configures the pin type and interrupt behavior.

Offset:0x0C01004C
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = GPIO<br>■ 1 = I2S_MCK (digital audio master clock in/out)<br>■ 2:3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:9 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 9:<br>■ 0 = Interrupt disabled for GPIO 9.<br>■ 1 = Interrupt on 0 → edge of GPIO pin 9<br>■ 2 = Interrupt on 1 → edge of GPIO pin 9<br>■ 3 = Interrupt on any edge of GPIO pin 9<br>■ 4 = Interrupt on level 0 of GPIO pin 9<br>■ 5 = Interrupt on level 1 of GPIO pin 9<br>■ 6:9 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 9 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 9<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.21 GPIO 10 Configuration (GPIO_PIN10)*

Configures the pin type and interrupt behavior.

Offset:0x0C010050
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = GPIO<br>■ 1 = I2S_MCK (digital audio master clock in/out)<br>■ 2:3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:10 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 10:<br>■ 0 = Interrupt disabled for GPIO 10.<br>■ 1 = Interrupt on 0 → edge of GPIO pin 10<br>■ 2 = Interrupt on 1 → edge of GPIO pin 10<br>■ 3 = Interrupt on any edge of GPIO pin 10<br>■ 4 = Interrupt on level 0 of GPIO pin 10<br>■ 5 = Interrupt on level 1 of GPIO pin 10<br>■ 6:10 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 10 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 10<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.22 GPIO 11 Configuration (GPIO_PIN11)*

Configures the pin type and interrupt behavior.

Offset:0x0C010054
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = GPIO<br>■ 1 = I2S_MCK (digital audio master clock out)<br>■ 2:3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:11 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 11:<br>■ 0 = Interrupt disabled for GPIO 11.<br>■ 1 = Interrupt on 0 → edge of GPIO pin 11<br>■ 2 = Interrupt on 1 → edge of GPIO pin 11<br>■ 3 = Interrupt on any edge of GPIO pin 11<br>■ 4 = Interrupt on level 0 of GPIO pin 11<br>■ 5 = Interrupt on level 1 of GPIO pin 11<br>■ 6:11 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 11 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 11<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.23 GPIO 12 Configuration (GPIO_PIN12)*

Configures the pin type and interrupt behavior.

Offset:0x0C010058
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = LB_DATA_8<br>■ 1 = SPI_CS<br>■ 2 = UART_CTS<br>■ 3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:12 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 12:<br>■ 0 = Interrupt disabled for GPIO 12.<br>■ 1 = Interrupt on 0 → edge of GPIO pin 12<br>■ 2 = Interrupt on 1 → edge of GPIO pin 12<br>■ 3 = Interrupt on any edge of GPIO pin 12<br>■ 4 = Interrupt on level 0 of GPIO pin 12<br>■ 5 = Interrupt on level 1 of GPIO pin 12<br>■ 6:12 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 12 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 12<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.24 GPIO 13 Configuration (GPIO_PIN13)*

Configures the pin type and interrupt behavior.

Offset:0x0C01005C
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = LB_WAIT_L<br>■ 1 = SPI_MOSI<br>■ 2 = UART_RTS<br>■ 3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:13 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 13:<br>■ 0 = Interrupt disabled for GPIO 13.<br>■ 1 = Interrupt on 0 → edge of GPIO pin 13<br>■ 2 = Interrupt on 1 → edge of GPIO pin 13<br>■ 3 = Interrupt on any edge of GPIO pin 13<br>■ 4 = Interrupt on level 0 of GPIO pin 13<br>■ 5 = Interrupt on level 1 of GPIO pin 13<br>■ 6:13 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 13 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 13<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.25 GPIO 14 Configuration (GPIO_PIN14)*

Configures the pin type and interrupt behavior.

Offset:0x0C010060
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = LB_INT_L<br>■ 1 = GPIO<br>■ 2 = UART_CLK (optional external UART clock)<br>■ 3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:14 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 14:<br>■ 0 = Interrupt disabled for GPIO 14.<br>■ 1 = Interrupt on 0 → edge of GPIO pin 14<br>■ 2 = Interrupt on 1 → edge of GPIO pin 14<br>■ 3 = Interrupt on any edge of GPIO pin 14<br>■ 4 = Interrupt on level 0 of GPIO pin 14<br>■ 5 = Interrupt on level 1 of GPIO pin 14<br>■ 6:14 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 14 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 14<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.26 GPIO 15 Configuration (GPIO_PIN15)*

Configures the pin type and interrupt behavior.

Offset:0x0C010064
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = LB_DATA_5<br>■ 1 = GPIO<br>■ 2:3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:15 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 15:<br>■ 0 = Interrupt disabled for GPIO 15.<br>■ 1 = Interrupt on 0 → edge of GPIO pin 15<br>■ 2 = Interrupt on 1 → edge of GPIO pin 15<br>■ 3 = Interrupt on any edge of GPIO pin 15<br>■ 4 = Interrupt on level 0 of GPIO pin 15<br>■ 5 = Interrupt on level 1 of GPIO pin 15<br>■ 6:15 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 15 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 15<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.27 GPIO 16 Configuration (GPIO_PIN16)*

Configures the pin type and interrupt behavior.

Offset:0x0C010068
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = LB_DATA_6<br>■ 1 = GPIO<br>■ 2:3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:16 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 16:<br>■ 0 = Interrupt disabled for GPIO 16.<br>■ 1 = Interrupt on 0 → edge of GPIO pin 16<br>■ 2 = Interrupt on 1 → edge of GPIO pin 16<br>■ 3 = Interrupt on any edge of GPIO pin 16<br>■ 4 = Interrupt on level 0 of GPIO pin 16<br>■ 5 = Interrupt on level 1 of GPIO pin 16<br>■ 6:16 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 16 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 16<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.28 GPIO 17 Configuration (GPIO_PIN17)*

Configures the pin type and interrupt behavior.

Offset:0x0C01006C
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:11 | CONFIG | Configures the function of the GPIO pin. Pins functions are independently selected. See "BGA Pin Descriptions" for function details.<br>■ 0 = LB_DATA_7<br>■ 1 = GPIO<br>■ 2:3 = Reserved |
| 10 | WAKEUP_ENABLE | When set, an interrupt caused by this GPIO pin is also sent to the power control FSM, causing interrupt events on this pin to wake the chip if in SLEEP mode.<br>Wakeup enable interrupt pulses must be at least 2 ms in width to ensure they are captured by the high speed clock after wakeup. Wakeup enable interrupts can only be level sensitive, not edge sensitive.<br>■ 0 = Interrupt on this pin does not cause SLEEP wakeup<br>■ 1 = Interrupt on this pin causes SLEEP wakeup |
| 9:17 | INT_TYPE | An interrupt will be set if the following occurs on GPIO pin 17:<br>■ 0 = Interrupt disabled for GPIO 17.<br>■ 1 = Interrupt on 0 → edge of GPIO pin 17<br>■ 2 = Interrupt on 1 → edge of GPIO pin 17<br>■ 3 = Interrupt on any edge of GPIO pin 17<br>■ 4 = Interrupt on level 0 of GPIO pin 17<br>■ 5 = Interrupt on level 1 of GPIO pin 17<br>■ 6:17 = Reserved |
| 6:3 | RES | Reserved |
| 2 | PAD_DRIVER | GPIO Pin 17 driver type.<br>■ 0 = Push/pull driver<br>■ 1 = Open drain driver |
| 1 | RES | Reserved |
| 0 | SOURCE | Output source for GPIO pin 17<br>■ 0 = GPIO_OUT register<br>■ 1 = Sigma delta PWM resource 0 |

### *7.5.29 SDIO Pin Driver Configuration (SDIO_PIN)*

Configures the SDIO pins.

Offset:0x0C010070
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:4 | RES | Reserved |
| 3:2 | PAD_PULL | SDIO data pin pad pull.<br>■ 0 = No pull<br>■ 1 = Pull-Up<br>■ 2 = Pull-Down<br>■ 3 = Reserved |
| 1:0 | PAD_STRENGTH | SDIO data pin drive strength.<br>■ 0 = 6 mA driver<br>■ 1 = 12 mA driver<br>■ 2 = 16 mA driver<br>■ 3 = 24 mA driver |

### *7.5.30 CLK_REQ Pin Driver Configuration (CLK_REQ_PIN)*

Configures the CLK_REQ pin.

Offset:0x0C010074
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:5 | RES | Reserved |
| 4 | OEN | CLK_REQ Output enable. This pin must be enabled if an external clock source relies on CLK_REQ to drive a clock.<br>■ 0 = CLK_REQ pin output enable is asserted<br>■ 1 = CLK_REQ pin output enable is not asserted |
| 3:2 | PAD_PULL | CLK_REQ data pin pad pull.<br>■ 0 = No pull<br>■ 1 = Pull-up<br>■ 2 = Pull-down<br>■ 3 = Reserved |
| 1:0 | PAD_STRENGTH | CLK_REQ data pin drive strength.<br>■ 0 = 6 mA driver<br>■ 1 = 12 mA driver<br>■ 2 = 16 mA driver<br>■ 3 = 23 mA driver |

### 7.5.31 Sigma Delta PWM Configuration (SIGMA_DELTA)

Offset:0x0C010078
Reset Value: 0x0
Access: Read/Write

Configures the output waveform for the sigma delta pulse waveform modulator (PWM), which can be used as a low frequency DAC. The maximum frequency that the target value can change for accurate DAC results is 256 * PRESCALAR.

| Bits | Bit Name | Description |
|---|---|---|
| 31:17 | RES | Reserved |
| 16 | ENABLE | Enables the PWM. The PWM should be disabled when not used to save power. |
| 15:8 | PRESCALAR | The clock source fed into the PWM will be divided by the PRESCALAR.<br>■ 0 = Use undivided clock source.<br>■ 1 = Divide clock by 2<br>■ 2 = Divide clock by 3<br>...<br>■ 255 = Divide clock by 256 |
| 7:0 | TARGET | Target value of the PWM. The output bitstream is asserted for TARGET/256 clocks, on average. |

## 7.6 AR6001X Side MBOX and Host IF Registers

Table 7-6 summarizes the AR6001X-side mailbox and host interface registers used by the AR6001X CPU and AHB.

*Table 7-6.* **AR6001-Side MBOX and Host IF Registers**

| Offset | Name | Description | Page |
|---|---|---|---|
| 0x0C014000 | MBOX_FIFO | Mailbox PIO access | page 120 |
| 0x0C014010 | MBOX_FIFO_STATUS | Non-destructive FIF status query | page 121 |
| 0x0C014014 | MBOX_DMA_POLICY | Mailbox DMA engine policy control | page 121 |
| 0x0C014018 | MBOX0_DMA_RX_DESCRIPTOR_BASE | Mailbox 0 Rx DMA descriptor address | page 122 |
| 0x0C01401C | MBOX0_DMA_RX_CONTROL | Mailbox 0 Rx DMA control | page 122 |
| 0x0C014020 | MBOX0_DMA_TX_DESCRIPTOR_BASE | Mailbox 0 Tx DMA descriptor address | page 123 |
| 0x0C014024 | MBOX0_DMA_TX_CONTROL | Mailbox 0 Tx DMA control | page 123 |
| 0x0C014028 | MBOX1_DMA_RX_DESCRIPTOR_BASE | Mailbox 1 Rx DMA descriptor address | page 123 |
| 0x0C01402C | MBOX1_DMA_RX_CONTROL | Mailbox 1 Rx DMA control | page 123 |
| 0x0C014030 | MBOX1_DMA_TX_DESCRIPTOR_BASE | Mailbox 1 Tx DMA descriptor address | page 124 |
| 0x0C014034 | MBOX1_DMA_TX_CONTROL | Mailbox 1 Tx DMA control | page 124 |
| 0x0C014038 | MBOX2_DMA_RX_DESCRIPTOR_BASE | Mailbox 2 Rx DMA descriptor address | page 124 |
| 0x0C01403C | MBOX2_DMA_RX_CONTROL | Mailbox 2 Rx DMA control | page 124 |
| 0x0C014040 | MBOX2_DMA_TX_DESCRIPTOR_BASE | Mailbox 2 Tx DMA descriptor address | page 124 |
| 0x0C014044 | MBOX2_DMA_TX_CONTROL | Mailbox 2 Tx DMA control | page 125 |
| 0x0C014048 | MBOX3_DMA_RX_DESCRIPTOR_BASE | Mailbox 3 Rx DMA descriptor address | page 125 |
| 0x0C01404C | MBOX3_DMA_RX_CONTROL | Mailbox 3 Rx DMA control | page 125 |
| 0x0C014050 | MBOX3_DMA_TX_DESCRIPTOR_BASE | Mailbox 3 Tx DMA descriptor address | page 126 |
| 0x0C014054 | MBOX3_DMA_TX_CONTROL | Mailbox 3 Tx DMA control | page 126 |
| 0x0C014058 | MBOX_INT_STATUS | Mailbox-related interrupt status | page 126 |
| 0x0C01405C | MBOX_INT_ENABLE | Mailbox-related interrupt enable | page 127 |
| 0x0C014060 | INT_HOST | Host CPU interrupt | page 128 |
| 0x0C014080 | LOCAL_COUNT | Credit counter direct access | page 128 |
| 0x0C0140A0 | COUNT_INC | Credit counter atomic increment | page 128 |
| 0x0C0140C0 | LOCAL_SCRATCH | Interface scratch | page 128 |
| 0x0C0140E0 | USE_LOCAL_BUS | Local bus configuration | page 129 |
| 0x0C0140E4 | SDIO_CONFIG | SDIO configuration | page 129 |
| 0x0C0140EC | STEREO_CONFIG | Stereo block configuration | page 129 |
| 0x0C0140F0 | STEREO_VOLUME | Set stereo volume | page 130 |
| 0x0C016000 | HOST_IF_WINDOW | Host interface access | page 131 |

Table 7-7 lists the AR6001X host interface data registers.

*Table 7-7.* **Host Interface Data Registers**

| Register Name | Address | Description |
|---|---|---|
| MBOX0 | 0x000–0xFE | Writing to any byte in this range pushes the data byte onto the MBOX Tx FIFO. Reading from any byte in this range pops data from the MBOX Rx FIFO. |
| MBOX0_EOM | 0xFF | Writing to any byte in this range pushes the data byte onto the MBOX Tx FIFO. Reading from any byte in this range pops data from the MBOX Rx FIFO.<br>If the MBOX0_EOM bit is set in the MBOX_CONFIG register, a write also sets EOM marker for this data |
| MBOX1 | 0x100–0x1FE | Writing to any byte in this range pushes the data byte onto the MBOX Tx FIFO. Reading from any byte in this range pops data from the MBOX Rx FIFO. |
| MBOX1_EOM | 0x1FF | |
| MBOX2 | 0x200–0x2FE | |
| MBOX2_EOM | 0x2FF | |
| MBOX3 | 0x300–0x3FE | |
| MBOX3_EOM | 0x3FF | |
| Control Register | 0x400–0x5FF | Control register address space |
| MBOX0 | 0x800–0x9FE | Alias to MBOX0 space. This alias behaves the same as MBOX0, but provides a larger addressing window for the SDIO and SPI interfaces. |
| MBOX0_EOM | 0x9FF | Alias to MBOX0_EOM |
| MBOX1 | 0x1000–17FE | Writing to any byte in this range pushes the data byte onto the MBOX Tx FIFO. Reading from any byte in this range pops data from the MBOX Rx FIFO. |
| MBOX1_EOM | 0x17FF | |
| MBOX2 | 0x1800–0x1FFE | |
| MBOX2_EOM | 0x1FFF | |
| MBOX3 | 0x2000–0x27FE | |
| MBOX3_EOM | 0x27FF | |

### *7.6.1 MBOX PIO Access (MBOX_FIFO)*

Offset:0x0C014000
Reset Value: 0x0
Access: Read/Write

This register provides PIO access to the mailbox FIFOs. An individual mailbox should be accessed over PIO or DMA, accessing the same mailbox with both the PIO and DMA causes undefined results.

Data can be written (FIFO push) or read (FIFO pop) one byte at a time using these FIFO PIO registers. Each mailbox is accessed according to the word address offset within this array:

- Address[3:0] = 0x0 accesses MBOX0
- Address[3:0] = 0x4 accesses MBOX1
- Address[3:0] = 0x8 accesses MBOX2
- Address[3:0] = 0xC accesses MBOX3

The data LSB for reads and writes contains FIFO data. Status bits for all mailboxes are always returned on every read, regardless of which mailbox is being popped. Status bits are provided so software may read this register without first reading MBOX_FIFO_STATUS, as a read to an empty FIFO is non-descructive. The status bit format in this register is identical to the format to the MBOX_FIFO_STATUS register, see the below register description for more details.

- PIO Read: Pops one byte of data from the Tx FIFO head
- PIO Write: Pushes one byte data onto the Rx FIFO tail
- Register Format:

| | |
|---|---|
| [31:20] | Reserved |
| EMPTY [19:16] | Read only status, Tx FIFO Empty |
| [15:12] | Read only status, Rx FIFO Full |
| [11:9] | Reserved |
| EOM[8] | Returns EOM on read, sets EOM bit for the mailbox addressed on write |
| DATA [7:0] | Returns data at the head of the FIFO. If the EMPTY bit is set for the mailbox which was read, the pop will not occur and DATA is not valid. If the FIFO is full during a write, the push will not occur. |

### 7.6.2 Non-Destructive FIFO Status Query (MBOX_FIFO_STATUS)

Offset:0x0C014010
Reset Value: 0x0
Access: Read only

This register returns the status of the mailbox FIFOs. It is has the same format as the MBOX_FIFO registers except that it does not contain the DATA field. This register may be read at any time without changing the mailbox state.

| Bits | Bit Name | Description |
|---|---|---|
| 31:20 | RES | Reserved |
| 19:16 | EMPTY | Read: Returns Empty status for the Tx mailbox.<br>■ bit 3 = MBOX 3 Tx FIFO is EMPTY<br>■ bit 2 = MBOX 2 Tx FIFO is EMPTY<br>■ bit 1 = MBOX 1 Tx FIFO is EMPTY<br>■ bit 0 = MBOX 0 Tx FIFO is EMPTY |
| 15:12 | FULL | Read: Returns Full status for the Rx mailbox.<br>■ bit 3 = MBOX 3 Rx FIFO is FULL<br>■ bit 2 = MBOX 2 Rx FIFO is FULL<br>■ bit 1 = MBOX 1 Rx FIFO is FULL<br>■ bit 0 = MBOX 0 Rx FIFO is FULL |
| 11:0 | RES | Reserved |

### 7.6.3 MBOX DMA Engine Policy Control (MBOX_DMA_POLICY)

Offset:0x0C014014
Reset Value: 0x0
Access: Read/Write

This register controls the queue service policy of the mailbox DMA engines. The Rx and Tx engines can be programmed independently to service their queues in round-robin or strict-priority order. The engines can also be programmed to make a new queue choice at the end of messages or individual descriptors. The default mode is round-robin decisions being made at the end of each message.

| Bits | Bit Name | Description |
|---|---|---|
| 31:4 | RES | Reserved |
| 3 | TXQUANTUM | Programming this field to a zero forces the Tx mailbox DMA engine to make queue service choices only at the end of messages (i.e. upon completing descriptors with the EOM bit set), while programming to a one allows it to make choices upon the completion of every descriptor. |
| 2 | TXORDER | Programming this field to a zero chooses round-robin and programming to a one chooses strict-priority (queue 0 is the highest priority) service ordering of mailbox Tx queues. |
| 1 | RXQUANTUM | Programming this field to a zero forces the Rx mailbox DMA engine to make queue service choices only at the end of messages (i.e. upon completing descriptors with the EOM bit set), while programming to a one allows it to make choices upon the completion of every descriptor. |
| 0 | RXORDER | Programming this field to a zero chooses round-robin and programming to a one chooses strict-priority (queue 0 is the highest priority) service ordering of mailbox Rx queues. |

### 7.6.4 MBOX 0 Rx DMA Descriptor Base Address (MBOX0_DMA_RX_DESCRIPTOR_BASE)

Offset:0x0C014018
Reset Value: 0x0
Access: Read/Write

This register holds the starting address of the descriptor chain for mailbox 0's Rx direction transfers. The DMA engine starts by fetching a descriptor from this address when the START bit in the "MBOX 0 Rx DMA Control (MBOX0_DMA_RX_CONTROL)" register is set. All DMA descriptors must be four-byte aligned, so the bottom two bits of this register's contents, as well as the bottom two bits of the next descriptor field of the individual descriptors are ignored and assumed to be zeros by the DMA engine. For the purposes of the DMA engine, Rx direction is defined to be transfers from the AR6001X to the host interface (nominally, data received from the antenna). The Tx direction is defined to be transfers from the host interface to AR6001X (nominally, data to transmit to the antenna).

| Bits | Bit Name | Description |
|---|---|---|
| 31:28 | RES | Reserved |
| 27:2 | ADDRESS | Most significant 26 bits of the four-byte-aligned address of the first descriptor in the DMA chain |
| 1:0 | RES | Reserved |

### 7.6.5 MBOX 0 Rx DMA Control (MBOX0_DMA_RX_CONTROL)

Offset:0x0C01401C
Reset Value: 0x0
Access: Read/Write

This register controls the operational state of the DMA engine for mailbox 0's Rx direction transfers. The register should always be written in a manner such that only one of the operations should be specified, and can be polled to see if the desired operation has taken effect (indicated by clearing the corresponding bit). The DMA engine starts out stopped and must always be kicked off for the first time with a START operation. The START operation causes the DMA engine to start fetching a descriptor at the address specified by the "MBOX 0 Rx DMA Descriptor Base Address (MBOX0_DMA_RX_DESCRIPTOR_BASE)" registers. Once this first descriptor is fetched, if the DMA engine ever catches up with a CPU-owned descriptor, it can be requested to refetch the descriptor that it stalled on by programming the RESUME operation. Software can stop DMA engine operation by programming the STOP operation. When the STOP operation is programmed, the DMA engine stops transfers immediately if it was already idle or at the end of the transfer of the current descriptor it's working on if it was busy. Note that this may leave incomplete messages in the mailbox FIFOs if the message in progress is scattered or gathered across multiple descriptors.

| Bits | Bit Name | Description |
|---|---|---|
| 31:3 | RES | Reserved |
| 2 | RESUME | Programming a one to this field causes a potentially stalled (due to having caught up with CPU-owned descriptors) DMA engine to resume transfers by refetching the last descriptor it had fetched and found to be CPU-owned. Software can use RESUME operations to keep adding descriptors to the end of the descriptor chain (only modifying CPU-owned descriptors) in a race-free atomic manner. If the RESUME operation is programmed and the DMA engine is not stalled, it has no effect and is automatically cleared. |
| 1 | START | Programming a one to this field causes the DMA engine to start transferring data by fetching the descriptor pointed to by the register MBOX0_DMA_RX_DESCRIPTOR_BASE. The START operation should usually be used only when the DMA engine is known to be stopped (after power-on or SOC reset) or after an explicit STOP operation. |
| 0 | STOP | Programming a one to this field causes the DMA engine to stop transferring more data from this descriptor chain after the current descriptor is completed, if a transfer is already in progress. |

### 7.6.6 MBOX 0 Tx DMA Descriptor Base Address (MBOX0_DMA_TX_DESCRIPTOR_BASE)

Offset:0x0C014020
Reset Value: 0x0
Access: Read/Write

Refer to the register "MBOX 0 Rx DMA Descriptor Base Address (MBOX0_DMA_RX_DESCRIPTOR_BASE)" on page 122 for a description.

| Bits | Bit Name | Description |
|---|---|---|
| 31:28 | RES | Reserved |
| 27:2 | ADDRESS | Programming a one to this field causes the DMA engine to start transferring data by fetching the descriptor pointed to by the MBOX0_DMA_RX_DESCRIPTOR_BASE register<br>The START operation should usually be used only when the DMA engine is known to be stopped (after power-on or SOC reset) or after an explicit STOP operation. |
| 1:0 | RES | Reserved |

### 7.6.7 MBOX 0 Tx DMA Control (MBOX0_DMA_TX_CONTROL)

Offset:0x0C014024
Reset Value: 0x0
Access: Read/Write

Refer to the register "MBOX 0 Rx DMA Descriptor Base Address (MBOX0_DMA_RX_DESCRIPTOR_BASE)" on page 122 for a description.

| Bits | Bit Name | Description |
|---|---|---|
| 31:3 | RES | Reserved |
| 2 | RESUME | See MBOX0_DMA_RX_CONTROL for details |
| 1 | START | See MBOX0_DMA_RX_CONTROL for details |
| 0 | STOP | See MBOX0_DMA_RX_CONTROL for details |

### 7.6.8 MBOX 1 Rx DMA Descriptor Base Address (MBOX1_DMA_RX_DESCRIPTOR_BASE)

Offset:0x0C014028
Reset Value: 0x0
Access: Read/Write

Refer to the register "MBOX 0 Rx DMA Control (MBOX0_DMA_RX_CONTROL)" on page 122 for a description.

| Bits | Bit Name | Description |
|---|---|---|
| 31:28 | RES | Reserved |
| 27:2 | ADDRESS | See MBOX0_DMA_RX_DESCRIPTOR_BASE for details |
| 1:0 | RES | Reserved |

### 7.6.9 MBOX 1 Rx DMA Control (MBOX1_DMA_RX_CONTROL)

Offset:0x0C01402C
Reset Value: 0x0
Access: Read/Write

Refer to the register "MBOX 0 Rx DMA Control (MBOX0_DMA_RX_CONTROL)" on page 122 for a description.

| Bits | Bit Name | Description |
|---|---|---|
| 31:3 | RES | Reserved |
| 2 | RESUME | See MBOX0_DMA_RX_CONTROL for details |
| 1 | START | See MBOX0_DMA_RX_CONTROL for details |
| 0 | STOP | See MBOX0_DMA_RX_CONTROL for details |

### 7.6.10 MBOX 1 Tx DMA Descriptor Base Address (MBOX1_DMA_TX_DESCRIPTOR_BASE)

Offset:0x0C014030
Reset Value: 0x0
Access: Read/Write

See the description of "MBOX 0 Rx DMA Descriptor Base Address (MBOX0_DMA_RX_DESCRIPTOR_BASE)" on page 122, as applied to mailbox 1's Tx direction transfers.

| Bits | Bit Name | Description |
|---|---|---|
| 31:28 | RES | Reserved |
| 27:2 | ADDRESS | See MBOX0_DMA_RX_DESCRIPTOR_BASE for details |
| 1:0 | RES | Reserved |

### 7.6.11 MBOX 1 Tx DMA Control (MBOX1_DMA_TX_CONTROL)

Offset:0x0C014034
Reset Value: 0x0
Access: Read/Write

See the description of "MBOX 0 Rx DMA Control (MBOX0_DMA_RX_CONTROL)" on page 122, as applied to mailbox 1's Tx direction transfers.

| Bits | Bit Name | Description |
|---|---|---|
| 31:3 | RES | Reserved |
| 2 | RESUME | See MBOX0_DMA_RX_CONTROL for details |
| 1 | START | See MBOX0_DMA_RX_CONTROL for details |
| 0 | STOP | See MBOX0_DMA_RX_CONTROL for details |

### 7.6.12 MBOX 2 Rx DMA Descriptor Base Address (MBOX2_DMA_RX_DESCRIPTOR_BASE)

Offset:0x0C014038
Reset Value: 0x0
Access: Read/Write

See the description of "MBOX 0 Rx DMA Descriptor Base Address (MBOX0_DMA_RX_DESCRIPTOR_BASE)" on page 122, as applied to mailbox 2's Rx direction transfers.

| Bits | Bit Name | Description |
|---|---|---|
| 31:28 | RES | Reserved |
| 27:2 | ADDRESS | See MBOX0_DMA_RX_DESCRIPTOR_BASE for details |
| 1:0 | RES | Reserved |

### 7.6.13 MBOX 2 Rx DMA Control (MBOX2_DMA_RX_CONTROL)

Offset:0x0C01403C
Reset Value: 0x0
Access: Read/Write

See the description of "MBOX 0 Rx DMA Control (MBOX0_DMA_RX_CONTROL)" on page 122, as applied to mailbox 2's Rx direction transfers.

| Bits | Bit Name | Description |
|---|---|---|
| 31:3 | RES | Reserved |
| 2 | RESUME | See MBOX0_DMA_RX_CONTROL for details |
| 1 | START | See MBOX0_DMA_RX_CONTROL for details |
| 0 | STOP | See MBOX0_DMA_RX_CONTROL for details |

### 7.6.14 MBOX 2 Tx DMA Descriptor Base Address (MBOX2_DMA_TX_DESCRIPTOR_BASE)

Offset:0x0C014040
Reset Value: 0x0
Access: Read/Write

See the description of "MBOX 0 Rx DMA Descriptor Base Address (MBOX0_DMA_RX_DESCRIPTOR_BASE)" on

page 122, as applied to mailbox 2's Tx direction transfers.

| Bits | Bit Name | Description |
|---|---|---|
| 31:28 | RES | Reserved |
| 27:2 | ADDRESS | See MBOX0_DMA_RX_DESCRIPTOR_BASE for details |
| 1:0 | RES | Reserved |

### *7.6.15 MBOX 2 Tx DMA Control (MBOX2_DMA_TX_CONTROL)*

Offset:0x0C014044
Reset Value: 0x0
Access: Read/Write

See the description of "MBOX 0 Rx DMA Control (MBOX0_DMA_RX_CONTROL)" on page 122, as applied to mailbox 2's Tx direction transfers.

| Bits | Bit Name | Description |
|---|---|---|
| 31:3 | RES | Reserved |
| 2 | RESUME | See MBOX0_DMA_RX_CONTROL for details |
| 1 | START | See MBOX0_DMA_RX_CONTROL for details |
| 0 | STOP | See MBOX0_DMA_RX_CONTROL for details |

### *7.6.16 MBOX 3 Rx DMA Descriptor Base Address (MBOX3_DMA_RX_DESCRIPTOR_BASE)*

Offset:0x0C014048
Reset Value: 0x0
Access: Read/Write

See the description of "MBOX 0 Rx DMA Descriptor Base Address (MBOX0_DMA_RX_DESCRIPTOR_BASE)" on page 122, as applied to mailbox 3's Rx direction transfers.

| Bits | Bit Name | Description |
|---|---|---|
| 31:28 | RES | Reserved |
| 27:2 | ADDRESS | See MBOX0_DMA_RX_DESCRIPTOR_BASE for details |
| 1:0 | RES | Reserved |

### *7.6.17 MBOX 3 Rx DMA Control (MBOX3_DMA_RX_CONTROL)*

Offset:0x0C01404C
Reset Value: 0x0
Access: Read/Write

See the description of "MBOX 0 Rx DMA Control (MBOX0_DMA_RX_CONTROL)" on page 122, as applied to mailbox 3's Rx direction transfers.

| Bits | Bit Name | Description |
|---|---|---|
| 31:3 | RES | Reserved |
| 2 | RESUME | See MBOX0_DMA_RX_CONTROL for details |
| 1 | START | See MBOX0_DMA_RX_CONTROL for details |
| 0 | STOP | See MBOX0_DMA_RX_CONTROL for details |

### 7.6.18 MBOX 3 Tx DMA Descriptor Base Address (MBOX3_DMA_TX_DESCRIPTOR_BASE)

Offset:0x0C014050
Reset Value: 0x0
Access: Read/Write

See the description of "MBOX 0 Rx DMA Descriptor Base Address (MBOX0_DMA_RX_DESCRIPTOR_BASE)" on page 122, as applied to mailbox 3's Tx direction transfers.

| Bits | Bit Name | Description |
|---|---|---|
| 31:28 | RES | Reserved |
| 27:2 | ADDRESS | See MBOX0_DMA_RX_DESCRIPTOR_BASE for details |
| 1:0 | RES | Reserved |

### 7.6.19 MBOX 3 Tx DMA Control (MBOX3_DMA_TX_CONTROL)

Offset:0x0C014054
Reset Value: 0x0
Access: Read/Write

See the description of "MBOX 0 Rx DMA Control (MBOX0_DMA_RX_CONTROL)" on page 122, as applied to mailbox 3's Tx direction transfers.

| Bits | Bit Name | Description |
|---|---|---|
| 31:3 | RES | Reserved |
| 2 | RESUME | See MBOX0_DMA_RX_CONTROL for details |
| 1 | START | See MBOX0_DMA_RX_CONTROL for details |
| 0 | STOP | See MBOX0_DMA_RX_CONTROL for details |

### 7.6.20 MBOX-Related Interrupt Status (MBOX_INT_STATUS)

Offset:0x0C014058
Reset Value: 0x0
Access: Read/Write

This register contains all AR6001 CPU interrupt sources associated with the mailbox and host interface.

| Bits | Bit Name | Description |
|---|---|---|
| 31:28 | RXDMA_COMPLETE | Per-mailbox Rx DMA completion (one descriptor completed) interrupts. Write one(s) to clear bit(s). |
| 27:24 | TXDMA_EOM_COMPLETE | Per-mailbox Tx DMA completion of end-of-message (descriptor with EOM flag completed) interrupts. Write one(s) to clear bit(s). |
| 23:20 | TXDMA_COMPLETE | Per-mailbox Tx DMA completion (one descriptor completed) interrupts. Write one(s) to clear bit(s). |
| 19:18 | RES | Reserved |
| 17 | TXOVERFLOW | MBOX Tx overflow error. The overflow condition is the same as the host interface overflow error. Write 1 to clear a bit. |
| 16 | RXUNDERFLOW | MBOX Rx underflow error. The underflow condition is the same as the host interface underflow error. Write 1 to clear a bit. |
| 15:12 | TXNOT_EMPTY | TXNOT_EMPTY pending interrupt for each of the 4 Tx mailboxes. Bit sets when there is no room in mbox FIFO. Write 1 to clear the bit.<br>■ Bit 3 = MBOX 3 TXNOT_EMPTY interrupt<br>■ Bit 2 = MBOX 2 TXNOT_EMPTY interrupt<br>■ Bit 1 = MBOX 1 TXNOT_EMPTY interrupt<br>■ Bit 0 = MBOX 0 TXNOT_EMPTY interrupt |

| Bits | Bit Name | Description |
|---|---|---|
| 11:8 | RXNOT_FULL | RXNOT_FULL pending interrupt for each Rx mailbox. Bit sets when there is 1 or more bytes in mbox FIFO. Write 1 to clear the bit.<br>■ Bit 3 = MBOX 3 RXNOT_FULL interrupt<br>■ Bit 2 = MBOX 2 RXNOT_FULL interrupt<br>■ Bit 1 = MBOX 1 RXNOT_FULL interrupt<br>■ Bit 0 = MBOX 0 RXNOT_FULL interrupt |
| 7:0 | HOST | Pending interrupt from host to AR6001X CPU. Write 1 to clear a bit.<br>■ Bit 7 = Interrupt #7<br>■ Bit 6 = Interrupt #6<br>■ Bit 5 = Interrupt #5<br>■ Bit 4 = Interrupt #4<br>■ Bit 3 = Interrupt #3<br>■ Bit 2 = Interrupt #2<br>■ Bit 1 = Interrupt #1<br>■ bit 0 = Interrupt #0 |

### *7.6.21 MBOX-Related Interrupt Enables (MBOX_INT_ENABLE)*

This register is used to mask/enable interrupts to the AR6001X CPU.

Offset:0x0C01405C
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:28 | RXDMA_COMPLETE | Enable per-mailbox Rx DMA completion interrupts |
| 27:24 | TXDMA_EOM_COMPLETE | Enable per-mailbox Tx DMA completion of end-of-message interrupts |
| 23:20 | TXDMA_COMPLETE | Enable per-mailbox Tx DMA completion interrupts |
| 19:18 | | Reserved |
| 17 | TXOVERFLOW | Enable MBOX Tx overflow error |
| 16 | RXUNDERFLOW | Enable MBOX Rx underflow error. |
| 15:12 | TXNOT_EMPTY | Enable TXNOT_EMPTY interrupts from mbox Tx FIFOs.<br>■ bit 3 = Enable MBOX 3 TXNOT_EMPTY interrupt<br>■ bit 2 = Enable MBOX 2 TXNOT_EMPTY interrupt<br>■ bit 1 = Enable MBOX 1 TXNOT_EMPTY interrupt<br>■ bit 0 = Enable MBOX 0 TXNOT_EMPTY interrupt |
| 11:8 | RXNOT_FULL | Enable RXNOT_FULL interrupts from mbox Rx FIFOs.<br>■ bit 3 = Enable MBOX 3 RXNOT_FULL interrupt<br>■ bit 2 = Enable MBOX 2 RXNOT_FULL interrupt<br>■ bit 1 = Enable MBOX 1 RXNOT_FULL interrupt<br>■ bit 0 = Enable MBOX 0 RXNOT_FULL interrupt |
| 7:0 | HOST | Enable pending interrupts from host to AR6001 CPU.<br>■ bit 7 = Enable Interrupt #7<br>■ bit 6 = Enable Interrupt #6<br>■ bit 5 = Enable Interrupt #5<br>■ bit 4 = Enable Interrupt #4<br>■ bit 3 = Enable Interrupt #3<br>■ bit 2 = Enable Interrupt #2<br>■ bit 1 = Enable Interrupt #1<br>■ bit 0 = Enable Interrupt #0 |

### 7.6.22 Host CPU Interrupt (INT_HOST)

Offset:0x0C014060
Reset Value: 0x0
Access: Read/Write

The AR6001X CPU may write to this register to interrupt to Host CPU. Software defines the meaning of each interrupt. Writes to this register set interrupt bits, the Host CPU must clear the bits. Note: this register is write 1 to set; write a 1 to each bit to be set. Writing a 0 does not change the bit value. These bits are cleared by hardware.

| Bits | Bit Name | Description |
|---|---|---|
| 31:8 | RES | Reserved |
| 7:0 | VECTOR | ■ Bit 7 = interrupt #7 (write 1 to set)<br>■ Bit 6 = interrupt #6 (write 1 to set)<br>■ Bit 5 = interrupt #5 (write 1 to set)<br>■ Bit 4 = interrupt #4 (write 1 to set)<br>■ Bit 3 = interrupt #3 (write 1 to set)<br>■ Bit 2 = interrupt #2 (write 1 to set)<br>■ Bit 1 = interrupt #1 (write 1 to set)<br>■ Bit 0 = interrupt #0 (write 1 to set) |

### 7.6.23 Credit Counters Direct Access (LOCAL_COUNT)

Offset:0x0C014080
Reset Value: 0x0
Access: Read/Write

Ordinary read/write access to credit counter registers. Read-modify-write operations are not atomic. Address decode is:

- Address[4:0] = 0x0 accesses COUNT0
- Address[4:0] = 0x4 accesses COUNT1
- Address[4:0] = 0x8 accesses COUNT2
- Address[4:0] = 0xc accesses COUNT3
- Address[4:0] = 0x10 accesses COUNT4
- Address[4:0] = 0x14 accesses COUNT5
- Address[4:0] = 0x18 accesses COUNT6
- Address[4:0] = 0x1C accesses COUNT7

Reset value for all counters is 0.

### 7.6.24 Credit Counter Atomic Increment (COUNT_INC)

Offset:0x0C0140A0
Reset Value: 0x0
Access: Read/Write

Reading or writing to this register cause a unit increment. Reads return the old value, then increment. Write data is ignored. Values of 0xFF do not increment.

- Address[4:0] = 0x0 increment COUNT0
- Address[4:0] = 0x4 increment COUNT1
- Address[4:0] = 0x8 increment COUNT2
- Address[4:0] = 0xC increment COUNT3
- Address[4:0] = 0x10 increment COUNT4
- Address[4:0] = 0x14 increment COUNT5
- Address[4:0] = 0x18 increment COUNT6
- Address[4:0] = 0x1C increment COUNT7

Reset value for all counters is 0.

### 7.6.25 Interface Scratch (LOCAL_SCRATCH)

Offset:0x0C0140C0
Reset Value: 0x0
Access: Read/Write

8 scratch registers are available for host and local CPU reading and writing. These registers are not atomic, data is always from the last writer.

- Address[4:0] = 0x0 access scratch register 0
- Address[4:0] = 0x4 access scratch register 1
- Address[4:0] = 0x8 access scratch register 2
- Address[4:0] = 0xC access scratch register 3
- Address[4:0] = 0x10 access scratch register 4
- Address[4:0] = 0x14 access scratch register 5
- Address[4:0] = 0x18 access scratch register 6
- Address[4:0] = 0x1C access scratch register 7

Reset value for all scratch registers is 0.

### 7.6.26 LB Configuration (USE_LOCAL_BUS)

Host interface function.

Offset:0x0C0140E0
Reset Value: 0x0
Access: Read only

| Bits | Bit Name | Description |
|---|---|---|
| 31:1 | RES | Reserved |
| 0 | PIN_INIT | ■ 0 = Board requests serial interface (SPI/SDIO)<br>■ 1 = Board requests system local bus/CF |

### 7.6.27 SDIO Configuration (SDIO_CONFIG)

The SDIO CCCR register is used by the host to probe basic state and functionality of the SDIO client card. This register reflects the state of relevant bits in the CCCR register.

Offset:0x0C0140E4
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:1 | RES | Reserved |
| 0 | CCCR_IOR1 | SDIO Function Ready 1. The AR6001X WLAN is mapped to SDIO function 1. After the local CPU has booted and is ready to communicate with the host CPU, this bit should be written to 1 to indicate readiness to the host. The host cannot ready function 1 registers until the bit is written. This register must be set to 1 before the timeout defined by the SDIO TPLFE_ENABLE_TIMEOUT_VAL. |

### 7.6.28 Stereo Block Configuration (STEREO_CONFIG)

This register controls the basic configuration of the stereo block.

Offset:0x0C0140EC
Reset Value: 0x0
Access: Read/Write

| Bits | Bit Name | Description |
|---|---|---|
| 31:26 | RES | Reserved |
| 25 | ENABLE | Enables the stereo block for operation |
| 24 | RESET | Resets the stereo buffers and $I^2S$ state. Hardware automatically clears to 0. |
| 23 | MIC_MASTER | Determines if microphone sample clock and WS come externally.<br>■ 0 = Use external MIC_SD, MIC_SCK<br>■ 1 = Use stereo clock |
| 22 | MIC_WORD_SIZE | Causes configures microphone word size:<br>■ 0 = 16-bit PCM words<br>■ 1 = 32-bit PCM words |
| 21:20 | MIC_MONO | Causes configures microphone stereo or mono.<br>■ 0 = Stereo<br>■ 1 = Mono from channel 0<br>■ 2 = Mono from channel 1<br>■ 3 = Reserved |
| 19:18 | STEREO_MONO | Causes configures stereo or mono.<br>■ 0 = Stereo<br>■ 1 = Mono from channel 0<br>■ 2 = Mono from channel 1<br>■ 3 = Reserved |
| 17:16 | DATA_WORD_SIZE | Controls the word size loaded into the PCM register from the mbox FIFO. Data word size:<br>■ 0 = Reserved<br>■ 1 = 16 bits/word<br>■ 2 = 24 bits/word<br>■ 3 = 32 bits/word |

| Bits | Bit Name | Description |
|---|---|---|
| 15 | I2S_WORD_SIZE | Controls the word size sent to the external $I^2S$ DAC. When set to 32 bit words, the PCM data is left-justified in the $I^2S$ word.<br>I2S word size:<br>■ 0 = 16 bits per $I^2S$ word<br>■ 1 = 32 bits per $I^2S$ word |
| 14 | MCK_SEL | When a DAC master clock is required, this field allows selects the raw clock source between refclk and the CORE_CLK.<br>■ 0 = Raw master clock is refclk (from crystal input)<br>■ 1 = Raw master clock is CORE_CLK (from CORE_CLK speed selection) |
| 13:10 | MCK_CNT | When a DAC master clock is required, this field allows the master clock divisor to be programmed. The clock selected by MCK_SEL is divided as follows:<br>■ 0 = Divide raw master clock by 2<br>■ 1 = Divide raw master clock by 4<br>■ 2 = Divide raw master clock by 6<br>...<br>■ 15 = Divide raw master clock by 32 |
| 9 | MCK_RAW | When a DAC master clock is required, this field allows the raw selected clock to be used.<br>■ 0 = Use master clock based on MCK_CNT toggles<br>■ 1 = Use raw master clock from MCK_SEL |
| 8 | MASTER | Controls the I2S_CK and I2S_WS master.<br>■ 0 = External DAC is the master and drives I2S_CK and I2S_WS<br>■ 1 = AR6001X is the master and drives I2S_CK and I2S_WS |
| 7:0 | POSEDGE | Controls timing between positive clock edges when the chip is in master mode. This number counts in units of refclk, which is the high speed input to the chip before the PLL. The time between positive edges of the stereo data clock defines the sample rate of the data.<br>This number can be calculated as follows:<br>POSEDGE = REFCLK_FREQ/(SAMPLE_RATE * WORD_SIZE * 2)<br>For example, a 32 kS/s sample rate with 16 bits/word and a 40 MHz refclk would yield:<br>POSEDGE = 40MHz/(32 KS/s * 16 bits/word * 2) = 39.06, round to 39. |

### *7.6.29 Set Stereo Volume (STEREO_VOLUME)*

Offset:0x0C0140F0
Reset Value: 0x0
Access: Read/Write

This register digitally attenuates or increases the volume level of the stereo output. Volume is adjusted in 3-db steps. If the gain is set too high, the PCM values saturate and waveform clipping occurs.

| Bits | Bit Name | Description |
|---|---|---|
| 31:13 | RES | Reserved |
| 12:8 | CHANNEL1 | Channel 1 gain/attenuation; a 5 bit number. The MSB is a sign bit, the others are magnitude. Binary (decimal): result<br>■ 11111 (–16): Maximum attenuation.<br>■ 11110 (–14): –84 db<br>...<br>■ 10001 (–1) : –6 db<br>■ 10000 (0) = : 0 db<br>■ 00000 (0) = : 0 db<br>■ 00001 (+1) : +6 db<br>...<br>■ 00111 (+7) : +42 db (maximum gain)<br>■ 01000 (+8) : Reserved<br>...<br>■ 01111 (+15): Reserved<br>Setting the gain above +7 is not supported. |

| Bits | Bit Name | Description |
|---|---|---|
| 7:5 | RES | Reserved |
| 4:0 | CHANNEL0 | Channel 0 gain/attenuation; a 5 bit number. The MSB is a sign bit, the others are magnitude. Binary (decimal): result<br>■ 11111 (–16): Maximum attenuation.<br>■ 11110 (–14): –84 db<br>...<br>■ 10001 (–1) : –6 db<br>■ 10000 (0) = : 0 db<br>■ 00000 (0) = : 0 db<br>■ 00001 (+1) : +6 db<br>...<br>■ 00111 (+7) : +42 db (maximum gain)<br>■ 01000 (+8) : Reserved<br>...<br>■ 01111 (+15): Reserved<br>Setting the gain above +7 is not supported. |

### *7.6.30 Host Interface Access (HOST_IF_WINDOW)*

Offset:0x0C016000
Reset Value: 0x0
Access: Read/Write

This register gives the AR6001 CPU access to the host interface address map. When the AR6001 CPU reads or writes this register, a transaction will be generated as is it came from the host interface. Since the host interface is byte addressed while the AR6001 address space is word addressed, software must shift the desired host interface address left by 2 to generate the proper window address.

This interface is intended for debug only, and may have unpredictable effects when window transactions conflict with host transactions

## *7.7 Host Interface Registers*

Table 7-8 summarizes the host interface registers, seen from the external host. Unlike the AHB registers, the address space is in bytes, not words. The CPU only views these registers using the HOST_IF_WINDOW register.

*Table 7-8.* **Host Interface Registers**

| Offset | Name | Description | Page |
|---|---|---|---|
| 0x00000400 | HOST_INT_STATUS | Address for AHB read access | page 133 |
| 0x00000401 | CPU_INT_STATUS | CPU-sourced interrupt status | page 133 |
| 0x00000402 | ERROR_INT_STATUS | Error or wakeup interrupt status | page 133 |
| 0x00000403 | COUNTER_INT_STATUS | Host interface credit counter interrupt | page 134 |
| 0x00000404 | MBOX_FRAME | Mailbox FIFO status | page 134 |
| 0x00000405 | RX_LOOKAHEAD_VALID | Valid bits for lookahead | page 134 |
| 0x00000408 | RX_LOOKAHEAD0 | Lookahead to next 4 MBOX Rx0 FIFO bytes | page 134 |
| 0x0000040C | RX_LOOKAHEAD1 | Lookahead to next 4 MBOX Rx1 FIFO bytes | page 136 |
| 0x00000410 | RX_LOOKAHEAD2 | Lookahead to next 4 MBOX Rx2 FIFO bytes | page 136 |
| 0x00000414 | RX_LOOKAHEAD3 | Lookahead to next 4 MBOX Rx3 FIFO bytes | page 136 |
| 0x00000418 | COUNT | Credit counter direct access | page 136 |
| 0x00000420 | COUNT_DEC | Credit counter atomic increment | page 136 |
| 0x00000440 | SCRATCH | Interface scratch | page 137 |
| 0x00000448 | INT_STATUS_ENABLE | HOST_INT_STATUS enable bits | page 137 |
| 0x00000449 | CPU_INT_STATUS_ENABLE | CPU-sourced interrupt status | page 137 |
| 0x0000044A | ERROR_STATUS_ENABLE | Error interrupt status | page 137 |
| 0x0000044b | COUNTER_INT_STATUS_ENABLE | Credit counter interrupt status | page 138 |
| 0x0000044C | FIFO_TIMEOUT | FIFO timeout period | page 138 |
| 0x0000044D | FIFO_TIMEOUT_ENABLE | FIFO timeout enable | page 138 |
| 0x0000044e | DISABLE_SLEEP | Disable sleep mode | page 138 |
| 0x00000460 | LOCAL_BUS_ENDIAN | Local bus Endianness | page 139 |
| 0x00000462 | LOCAL_BUS | Local bus and SPI host interface state | page 139 |
| 0x00000480 | SPI_CONFIG | SPI slave interface configuration | page 141 |

*Table 7-8.* **Host Interface Registers**

| Offset | Name | Description | Page |
|---|---|---|---|
| 0x00000481 | SPI_STATUS | SPI status | page 142 |
| 0x00000600 | CIS_WINDOW | SDIO CIS tuples copy | page 143 |

### 7.7.1 Pending Interrupt Status (HOST_INT_STATUS)

Offset:0x00000400
Reset Value: 0x0
Access: Read only

Reads to this register return pending host interrupt bits. Writes to this register clear interrupt bits. Note: Write a 1 to each bit to be cleared. All bits written as 0 do not update the interrupt status for that bit.

| Bits | Bit Name | Description |
|---|---|---|
| 7 | ERROR | Error or Wakeup Interrupt (read only, clear using ERROR_INT_STATUS register) |
| 6 | CPU | Interrupt from AR6001 CPU (read only, clear using CPU_INT_STATUS register) |
| 5 | AR6001_INT | Copy of the interrupt line to the AR6001 CPU. This interrupt is normally serviced by the AR6001 CPU, and should typically be disabled at the host interface. If the host does service this interrupt, more details can be read from the IN_STATUS register in the AR6001 address space, accessible via a window read. |
| 4 | COUNTER | Interrupt from software controlled credit counters. Read only, see the COUNTER_INT_STATUS register for details. |
| 3:0 | MBOX_DATA | Rx Data Pending in the corresponding MBOX (FIFO is not empty). This will be cleared when the FIFO is no longer empty. |

### 7.7.2 CPU-Sourced Interrupt Status (CPU_INT_STATUS)

Offset:0x00000401
Reset Value: 0x0
Access: Read/Write

Indicates the CPU interrupt condition being communicated by the AR6001X CPU. Software defines the meaning of each interrupt in this register. Writes to this register clear interrupt bits. Note: Write a 1 to each bit to be cleared. Writing a 0 does not change the bit value.

| Bits | Bit Name | Description |
|---|---|---|
| 7:0 | BIT | ■ Bit 7 = interrupt #7<br>■ Bit 6 = interrupt #6<br>■ Bit 5 = interrupt #5<br>■ Bit 4 = interrupt #4<br>■ Bit 3 = interrupt #3<br>■ Bit 2 = interrupt #2<br>■ Bit 1 = interrupt #1<br>■ Bit 0 = interrupt #0 |

### 7.7.3 Error or Wakeup Interrupt Status (ERROR_INT_STATUS)

Offset:0x00000402
Reset Value: 0x0
Access: Read/Write

Indicates a wakeup or error condition which caused the Error Interrupt in HOST_INT_STATUS.

| Bits | Bit Name | Description |
|---|---|---|
| 7:4 | RES | Reserved |
| 3 | SPI | SPI Error Interrupt. This error can only be masked or cleared be accessing the SPI-specific registers from the SPI host. |
| 2 | WAKEUP | The client transitioning to the ON state. Set as the client enters ON, and can immediately accept host transactions. Writing a 1 clears the register field. Writing a 0 does not change the bit value. |
| 1 | RXUNDERFLOW | The host attempted to read a mailbox which did not contain data, and no data was available for a timeout period. This implies a software flow control error. Writing a 1 clears the register field. Writing a 0 does not change the bit value. |
| 0 | TXOVERFLOW | The host attempted to write a mailbox which was full and had no available buffer space for a timeout period. This implies a software flow control error. Writing a 1 to clears the register field. Writing a 0 does not change the bit value. |

### 7.7.4 Host IF Credit Counter Interrupt (COUNTER_INT_STATUS)

Read-only register to return counter interrupt status.

Offset:0x00000403
Reset Value: 0x0
Access: Read only

| Bits | Bit Name | Description |
|---|---|---|
| 7:0 | COUNTER | Each counter sets and clears its interrupt bit as follows:<br>■ Set: Counter transitions from 0 → 1<br>■ Clear: Counter transitions from 1 → 0<br>Bit mapping is as follows:<br>■ Bit7 = Counter 7 interrupt<br>■ Bit6 = Counter 6 interrupt<br>■ Bit5 = Counter 5 interrupt<br>■ Bit4 = Counter 4 interrupt<br>■ Bit3 = Counter 3 interrupt<br>■ Bit2 = Counter 2 interrupt<br>■ Bit1 = Counter 1 interrupt<br>■ Bit0 = Counter 0 interrupt |

### 7.7.5 Mailbox FIFO Status (MBOX_FRAME)

This register returns current MBOX framing status.

Offset:0x00000404
Reset Value: 0x0
Access: Read only

| Bits | Bit Name | Description |
|---|---|---|
| 7:4 | RXEOM | Rx FIFO contains a data byte with EOM marker set in the corresponding mailbox |
| 3:0 | RXSOM | Rx FIFO contains a data byte with Start of Message (SOM) marker set in the corresponding mailbox. A SOM byte always follows an EOM byte from the previous message. |

### 7.7.6 Valid Bits for Lookahead (RX_LOOKAHEAD_VALID)

Read only register to return current lookahead valid bits. If the bit is set, all four lookahead bytes are valid. If the bit is cleared not all bytes are valid. If cleared but the MBOX is not empty, only the first byte is valid but the others are not.

Offset:0x00000405
Reset Value: 0x0
Access: Read only

| Bits | Bit Name | Description |
|---|---|---|
| 7:4 | RES | Reserved |
| 3:0 | MBOX | ■ Bit 3 = MBOX3 lookahead, all bytes valid<br>■ Bit 2 = MBOX2 lookahead, all bytes valid<br>■ Bit 1 = MBOX1 lookahead, all bytes valid<br>■ Bit 0 = MBOX0 lookahead, all bytes valid |

### 7.7.7 Lookahead to Next 4 MBOX Rx0 FIFO Bytes (RX_LOOKAHEAD0)

Offset:0x00000408
Reset Value: 0x0
Access: Read only

This read only register array returns the first 4 bytes of the MBOX Rx0 FIFO. Reading this register will not pop or otherwise change the contents of the FIFO, it is non-destructive.

- Address[1:0] = 0x0 returns MBOX0 Rx FIFO Head-3 byte
- Address[1:0] = 0x1 returns MBOX0 Rx FIFO Head-2 byte
- Address[1:0] = 0x2 returns MBOX0 Rx FIFO Head-1 byte
- Address[1:0] = 0x3 returns MBOX0 Rx FIFO Head byte

### 7.7.8 Lookahead to Next 4 MBOX Rx1 FIFO Bytes (RX_LOOKAHEAD1)

Offset:0x0000040C
Reset Value: 0x0
Access: Read only

This register array returns the first four bytes of the MBOX Rx1 FIFO. Reading this register not pop or otherwise change the contents of the FIFO, it is non-destructive.

- Address[1:0] = 0x0 returns MBOX1 Rx FIFO Head-3 byte
- Address[1:0] = 0x1 returns MBOX1 Rx FIFO Head-2 byte
- Address[1:0] = 0x2 returns MBOX1 Rx FIFO Head-1 byte
- Address[1:0] = 0x3 returns MBOX1 Rx FIFO Head byte

### 7.7.9 Lookahead to Next 4 MBOX Rx2 FIFO Bytes (RX_LOOKAHEAD2)

Offset:0x00000410
Reset Value: 0x0
Access: Read only

This read only register array returns the first four bytes of the MBOX Rx2 FIFO. Reading this register will not pop or otherwise change the contents of the FIFO, it is non-destructive.

- Address[1:0] = 0x0 returns MBOX2 Rx FIFO Head-3 byte
- Address[1:0] = 0x1 returns MBOX2 Rx FIFO Head-2 byte
- Address[1:0] = 0x2 returns MBOX2 Rx FIFO Head-1 byte
- Address[1:0] = 0x3 returns MBOX2 Rx FIFO Head byte

### 7.7.10 Lookahead to Next 4 MBOX Rx3 FIFO Bytes (RX_LOOKAHEAD3)

Offset:0x00000414
Reset Value: 0x0
Access: Read only

This read only register array returns the first four bytes of the MBOX Rx3 FIFO. Reading this register will not pop or otherwise change the contents of the FIFO, it is non-destructive.

- Address[1:0] = 0x0 returns MBOX3 Rx FIFO Head-3 byte
- Address[1:0] = 0x1 returns MBOX3 Rx FIFO Head-2 byte
- Address[1:0] = 0x2 returns MBOX3 Rx FIFO Head-1 byte
- Address[1:0] = 0x3 returns MBOX3 Rx FIFO Head byte

### 7.7.11 Credit Counters Direct Access (COUNT)

Offset:0x00000418
Reset Value: 0x0
Access: Read/Write

Ordinary read/write access to credit counter registers. Read-modify-write operations are not atomic. Address decode is:

- Address[2:0] = 0x7 accesses COUNT7
- Address[2:0] = 0x6 accesses COUNT6
- Address[2:0] = 0x5 accesses COUNT5
- Address[2:0] = 0x4 accesses COUNT4
- Address[2:0] = 0x3 accesses COUNT3
- Address[2:0] = 0x2 accesses COUNT2
- Address[2:0] = 0x1 accesses COUNT1
- Address[2:0] = 0x0 accesses COUNT0

Reset value for all counters is 0.

### 7.7.12 Credit Counter Atomic Decrement (COUNT_DEC)

Offset:0x00000420
Reset Value: 0x0
Access: Read/Write

Reading or writing to this register causes a unit decrement. Reads return the old value, then decrement. If the value of COUNT is 0, the decrement does not occur. If read returns a 0, software knows the decrement does not occur. Write data is ignored.

Registers are word aligned to allow 16-bit or 32-bit accesses from the host to read or write a single atomic register. Reads or writes to non-word aligned addresses have no effect.

- Address[4:0] == 0x1C decrement COUNT7
- Address[4:0] == 0x18 decrement COUNT6
- Address[4:0] == 0x14 decrement COUNT5
- Address[4:0] == 0x10 decrement COUNT4
- Address[4:0] == 0xc decrement COUNT3
- Address[4:0] == 0x8 decrement COUNT2
- Address[4:0] == 0x4 decrement COUNT1
- Address[4:0] == 0x0 decrement COUNT0

Reset value for all counters is 0.

### 7.7.13 Interface Scratch (SCRATCH)

Offset:0x00000440
Reset Value: 0x0
Access: Read/Write

Eight scratch registers are available for host and local CPU read/write. These registers are not atomic, data is always from the last writer.

- Address[2:0] == 0x0 scratch register 0
- Address[2:0] == 0x0 scratch register 0
- Address[2:0] == 0x0 scratch register 0
- Address[2:0] == 0x0 scratch register 0
- Address[2:0] == 0x0 scratch register 0
- Address[2:0] == 0x0 scratch register 0
- Address[2:0] == 0x0 scratch register 0

Reset value for all scratch registers is 0.

### 7.7.14 HOST_INT_STATUS Enable Bits (INT_STATUS_ENABLE)

Offset:0x00000448
Reset Value: 0x1
Access: Read/Write

Enable bits for the HOST_INT_STATUS register. Each bit enables the corresponding bit in the HOST_INT_STATUS register. Bit values are as follows:

- 0 = interrupt is disabled
- 1 = interrupt is enabled

| Bits | Bit Name | Description |
|---|---|---|
| 7 | ERROR | Enable Error Interrupt |
| 6 | CPU | Enable AR6001X CPU interrupt |
| 5 | AR6001_INT | Enable a copy of the AR6001X CPU interrupt to be sent to the host. This interrupt is normally serviced by the AR6001X CPU, and should typically be disabled at the host interface. |
| 4 | COUNTER | Enable counter interrupt |
| 3:0 | MBOX_DATA | Enable Rx Data Pending Interrupt in the corresponding MBOX |

### 7.7.15 CPU Sourced Interrupt Status (CPU_INT_STATUS_ENABLE)

Offset:0x00000449
Reset Value: 0x0
Access: Read/Write

Enable bits for the CPU_INT_STATUS register. Each bit enables the corresponding bit in the CPU_INT_STATUS register. Bit values are as follows:

- 0 = interrupt is disabled
- 1 = interrupt is enabled

| Bits | Bit Name | Description |
|---|---|---|
| 7:0 | BIT | ■ Bit 7 = enable interrupt #7<br>■ Bit 6 = enable interrupt #6<br>■ Bit 5 = enable interrupt #5<br>■ Bit 4 = enable interrupt #4<br>■ Bit 3 = enable interrupt #3<br>■ Bit 2 = enable interrupt #2<br>■ Bit 1 = enable interrupt #1<br>■ Bit 0 = enable interrupt #0 |

### 7.7.16 Error Interrupt Status (ERROR_STATUS_ENABLE)

Offset:0x0000044A
Reset Value: 0x0
Access: Read/Write

Enable bits for the ERROR_STATUS register. Each bit enables the corresponding bit in the ERROR_STATUS register.

- 0 = interrupt is disabled
- 1 = interrupt is enabled

| Bits | Bit Name | Description |
|---|---|---|
| 7:3 | RES | Reserved |
| 2 | WAKEUP | Wakeup interrupt enable |
| 1 | RXUNDERFLOW | RXUNDERFLOW interrupt enable |
| 0 | TXOVERFLOW | TXOVERFLOW interrupt enable |

### 7.7.17 Credit Counter Interrupt Status (COUNTER_INT_STATUS_ENABLE)

Offset:0x0000044B
Reset Value: 0x0
Access: Read/Write

Enable bits for the COUNTER_INT_STATUS register. Each bit enables the corresponding bit in the COUNTER_INT_STATUS register. Bit values are as follows:

- 0 = interrupt is disabled
- 1 = interrupt is enabled

| Bits | Bit Name | Description |
|---|---|---|
| 7:0 | BIT | ■ Bit 7 = Enable interrupt #7<br>■ Bit 6 = Enable interrupt #6<br>■ Bit 5 = Enable interrupt #5<br>■ Bit 4 = Enable interrupt #4<br>■ Bit 3 = Enable interrupt #3<br>■ Bit 2 = Enable interrupt #2<br>■ Bit 1 = Enable interrupt #1<br>■ Bit 0 = Enable interrupt #0 |

### 7.7.18 FIFO Timeout Period (FIFO_TIMEOUT)

Offset:0x0000044C
Reset Value: 0xFF
Access: Read/Write

If a MBOX Rx FIFO is empty and a host read arrives, or a MBOX Tx FIFO is full and a host write arrives, the SDIO interface wait for this timeout period before declaring an error state.

After an error is declared, all writes to full FIFOs are dropped, and all reads to empty FIFOs return garbage data (instead of waiting for data). After the error conditions have been cleared, a write to the ERROR_INT_STATUS clears the error condition and the FIFOs return to normal operation.

| Bits | Bit Name | Description |
|---|---|---|
| 7:0 | VALUE | Timeout value, in ms, when CORE_CLK=40 MHz, or in 0.5 ms when CORE_CLK=80 MHz. This bit should never be set to 0. |

### 7.7.19 FIFO Timeout Enable. (FIFO_TIMEOUT_ENABLE)

Offset:0x0000044D
Reset Value: 0x0
Access: Read/Write

This register can be used to disable all timeout conditions.

| Bits | Bit Name | Description |
|---|---|---|
| 7:1 | RES | Reserved |
| 0 | SET | ■ 1 = FIFO timeouts are enabled<br>■ 0 = FIFO timeouts are disabled |

### 7.7.20 Disable Sleep Mode (DISABLE_SLEEP)

Offset:0x0000044E
Reset Value: 0x0
Access: Read/Write

Disable AR6001X from entering SLEEP state.

| Bits | Bit Name | Description |
|---|---|---|
| 7:2 | RES | Reserved |
| 1 | FOR_INT | ■ 0 = AR6001 may enter SLEEP when a host interrupt is pending<br>■ 1 = AR6001 will never enter SLEEP when a host interrupt is pending |
| 0 | ON | ■ 0 = AR6001 may enter SLEEP state<br>■ 1 = AR6001 will never enter SLEEP state |

### 7.7.21 LB Endianness (LOCAL_BUS_ENDIAN)

Offset:0x00000460
Reset Value: 0x0
Access: Read/Write

Sets Endianness of host interface local bus. Internally, AR6001X is always Little Endian.

| Bits | Bit Name | Description |
|---|---|---|
| 7:1 | RES | Reserved |
| 0 | BIG | ■ 0 = Little Endian<br>■ 1 = Big Endian |

### 7.7.22 LB and SPI Host Interface State (LOCAL_BUS)

Offset:0x00000462
Reset Value: 0x0
SPI Host Address: 14'h0470
SPI Host Access: Read/Write
Access: Read only

In local bus and SPI modes, this register is used to probe the chip state. Unlike other registers, this one can be read and written when the chip is in sleep state. This register cannot be written by the AR6001 CPU, only by the host.

| Bits | Bit Name | Access | Reset Value | Description |
|---|---|---|---|---|
| 7:5 | RES | | 0x0 | Reserved |
| 4 | SOFT_RESET | R/W | 0x0 | Used by the host to reset the AR6001 core. Can be written and read by the host. |
| 3 | IO_ENABLE | R/W | 0x0 | ■ Local Bus: When set, LB_INT_L acts as an active-low level-sensitive interrupt. When the bit is cleared, LB_INT_L displays the ready status of the chip. Can be written by the host.<br>■ SPI: The host must write a 1 to IO_ENABLE before it can access the internal AR6001 registers. Writing a 1 will cause AR6001 to power up and boot. Writing a 0 will shut down AR6001. |
| 2 | KEEP_AWAKE | R/W | 0x1 | Forces AR6001X to stay awake. Can be used to bring AR6001 out of sleep without holding up the Local Bus interface. Can be written by the host. |
| 1:0 | STATE | RO | 0x0 | Returns the current chip state.<br>■ SHUTDOWN = 0<br>■ ON = 1<br>■ SLEEP = 2<br>■ WAKEUP = 3 |



### 7.7.23 AR6001X CPU Interrupt (INT_WLAN)

Offset:0x00000464
Reset Value: 0x0
Access: Read/Write

The host may write to this register to interrupt to AR6001 CPU. Software defines the meaning of each interrupt. Writes to this register will set interrupt bits, the AR6001X CPU must clear the bits.

Writing a 1 sets the register field. Writing a 0 does not change the bit value. These bits are cleared by hardware.

| Bits | Bit Name | Description |
|---|---|---|
| 7:0 | VECTOR | ■ Bit 7 = interrupt #7 (write 1 to set)<br>■ Bit 6 = interrupt #6 (write 1 to set)<br>■ Bit 5 = interrupt #5 (write 1 to set)<br>■ Bit 4 = interrupt #4 (write 1 to set)<br>■ Bit 3 = interrupt #3 (write 1 to set)<br>■ Bit 2 = interrupt #2 (write 1 to set)<br>■ Bit 1 = interrupt #1 (write 1 to set)<br>■ Bit 0 = interrupt #0 (write 1 to set) |

### *7.7.24 SPI Slave Interface Configuration (SPI_CONFIG)*

Offset:0x00000480
Reset Value: 0x0
Access: Read/Write

This register selects the data transfer size (8/16/32 bit), interrupt enable, and SPI SLAVE interface loopback test functionality. The RESET bit resets the SPI core. This register is accessible only from the SPI host side. It cannot be accessed using interfaces other that SPI.

| Bits | Bit Name | Access | Reset Value | Description |
|---|---|---|---|---|
| 7 | ENDIAN | R/W | 1 | Reflects Endianness of the operation register address and data.<br>■ 0 = Little Endian<br>■ 1 = Big Endian |
| 6 | RES | — | 0x0 | Reserved |
| 5 | SPI_CLK_OFFSET | R | Based on input pin | Reflects the clocking mode<br>■ 0 = No idle CLKS at the start of CS assertion<br>■ 1 = One idle CLK at the start of CS assertion |
| 4 | SPI_RESET | R/W | 0x0 | Controls the reset state of SPI interface.<br>■ 0 = Normal operational mode<br>■ 1 = Reset SPI core |
| 3 | INTERRUPT_ENABLE | R/W | 0x0 | Enables the SPI interface interrupt to propagate to AR6001 interrupt logic.<br>■ 0 = SPI interrupt disabled<br>■ 1 = SPI interrupt enabled |
| 2 | TEST_MODE | R/W | 0x0 | For test mode (Loopback) operation. When set, data received is transmitted back (echo) after 1 transaction delay.<br>■ 0 = SPI normal mode<br>■ 1 = SPI test mode |
| 1:0 | DATA_SIZE | R/W | 0x2 | Selects the data size for SPI. Note: The address phase is always 16-bit. (Default = 32-bit)<br>■ 0 = 8-bit data<br>■ 1 = 16-bit data<br>■ 2 = 32-bit data<br>■ 3 = Reserved |

*Detailed Description of the SPI_CONFIG Fields*

| Bit | Bit Name | Description |
|---|---|---|
| 7 | Endianness | Affects all Address phases and internal register reads and writes in 16- and 32-bit mode.<br>■ 1 = Big Endian<br>Example: 16 bit value = 16h80A5 (DATA16 mode)<br>Bit No 15 8 7 0<br>1 0 0 0 0 0 0 0 * 1 0 1 0 0 1 0 1<br>80 A5<br>Example: 32 bit value = 32h000080A5 (DATA32 mode)<br>Bit No 31 16 15 8 7 0<br>0 0 0 0 0 0 0 0 * 0 0 0 0 0 0 0 0 * 1 0 0 0 0 0 0 0 * 1 0 1 0 0 10 1<br>0 0 80 A5<br>■ 0 = Little Endian<br>Example: 32 bit value 32h000080A5 DATA32 mode<br>Bit No 31 16 15 8 7 0<br>1 0 1 0 0 1 0 1 * 1 0 0 0 0 0 0 0 * 0 0 0 0 0 0 0 0 * 0 0 0 0 0 0 0 0<br>A5 80 00 00 |
| 5 | Test Mode | 1 = Enter Test (Loop back) mode<br>This mode is for debug purposes only. This bit should be reset for normal operation. If set, all received bits on the SPI interface (on SPI_MOSI) are sent back to the Host on the SPI Interface (SPI_MISO). |
| 4 | Reset Mode | 1 = Assert active low reset<br>This mode results in the reset of all state machines of the SPI slave. All operation registers (config, status, address, and count registers) retain the last value. This bit is auto-clearing. |
| 3 | Interrupt Enable | ■ 0 = Interrupts are disabled<br>■ 1 = Interrupts are enabled, resulting in any error conditions (e.g., IF error, ADDR Error, RD Error, and WR error) to assert the INTR output of SPI_SLV |
| 2:0 | Data Size | ■ 00 = DATA8<br>All data phases are 8 bits in size<br>■ 01 = DATA16<br>All data phases are maximum 16 bits in size. Allowable data sizes are 16 bits for internal registers access, 8 and 16bits for mailbox single reads and writes, and 16 bits for DMA transfers. 8 bits are also allowed for the last data phase of a DMA transaction<br>■ 10 = DATA32<br>All data phases are maximum 32 bits in size. Allowable data sizes are 16-bit for internal registers access, 8-,16-, 24-, and 32-bit for mailbox single reads and writes, and 32-bit for DMA transfers. 8-,16-, and 24-bit is allowed for the last data phase of a DMA transaction |

### *7.7.25 SPI Status (SPI_STATUS)*

Offset:0x00000481
Reset Value: 0x0
SPI Host Address: 14'h0470
SPI Host Access: Read/Write
Access: Read/Write

This register indicates the status of the SPI core. Generally during any transaction, if SPI core (slave) has no data to transfer, contents of this register is shifted on the MISO pin. SPI host can use this data or ignore in not required.

| Bit | Bit Name | Access | Reset Value | Description |
|---|---|---|---|---|
| 7:6 | RES | | 0x0 | Reserved |
| 5 | DMA_OVER | R/WC | 0x0 | ■ 0 = No status<br>■ 1 = Last issued DMA is complete. Can be cleared by writing a 1 to this register. |

| Bit | Bit Name | Access | Reset Value | Description |
|---|---|---|---|---|
| 4 | IFF_ERR | | | ■ 0 = No error<br>■ 1 = Number of SPI CLKs received in a CS assertion was not a multiple of 8. Can be cleared by writing a 1 to this register. |
| 3 | ADDR_ERR | | | ■ 0 = No error<br>■ 1 = Non-existent internal register address received<br>OR: An 8- or 32-bit address/command phase received |
| 2 | RD_ERR | R/WC | 0x0 | ■ 0 = No error<br>■ 1 = Indicates read error occurred<br>Can be cleared by writing to this register. A read error is indicated by a 16- or 32- SPI_CLK data phase occurring in a DATA8 mode read, or by a 32- SPI_CLK data phase occurring in DATA16 mode. |
| 1 | WR_ERR | R/WC | 0x0 | ■ 0 = Indicates no write error<br>■ 1 = Indicates write error occurred<br>Can be cleared by writing a 1 to this register |
| 0 | READY | R | 0x1 | ■ 0 = Indicates current command pending<br>■ 1 = Indicates current request completed<br>Ready to accept SPI transaction |

### *7.7.26 SDIO CIS Tuples Copy (CIS_WINDOW)*

Offset:0x0C000600
Reset Value: 0x0
Access: Read/Write

This address space is a copy of the SDIO CIS tuples, the first tuple for CIS 0 begins at offset 0x0. The first tuple for CIS 1 (function 1 space) begins at offset 0x100. This space is read-only for general CIS access. CIS0 contains 32 bytes of programmable tuples, which begin after the last fixed tuple in CIS0. These tuples can be written by the MIPS CPU to pass configuration information to host drivers.

# 8. Package Dimensions

The AR6001X is packaged in BGA. It is a JEDEC MO-207 compliant 216 BGA package. The body size is 10 mm x 10 mm, and the ball pitch is 0.50 mm.

BGA package drawings are provided in Figure 8-1 and Table 8-1.

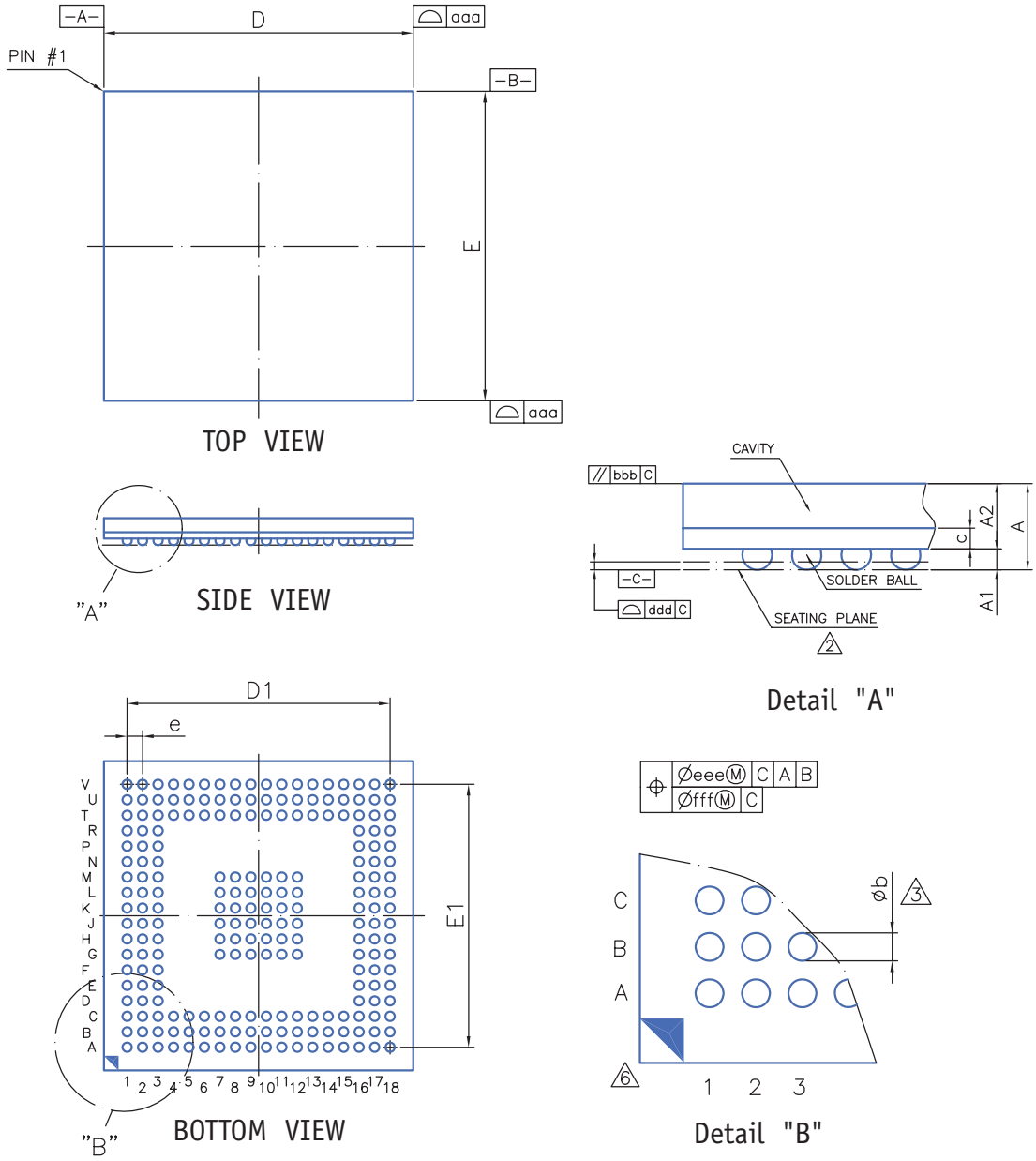



*Figure 8-1.* **BGA Package Drawing**

*Table 8-1.* **BGA Package Dimensions**

| Dimension Label | Min. | Nom. | Max. | Unit. | Min. | Nom. | Max. | Unit. |
|---|---|---|---|---|---|---|---|---|
| A | — | — | 1.00 | mm | — | — | 0.0039 | inches |
| A1 | 0.16 | 0.21 | 0.26 | mm | 0.006 | 0.008 | 0.010 | inches |
| A2 | 0.61 | 0.66 | 0.71 | mm | 0.024 | 0.026 | 0.028 | inches |
| c | 0.17 | 0.21 | 0.25 | mm | 0.007 | 0.008 | 0.010 | inches |
| D | 9.90 | 10.00 | 10.10 | mm | 0.390 | 0.394 | 0.398 | inches |
| E | 9.90 | 10.00 | 10.10 | mm | 0.390 | 0.394 | 0.398 | inches |
| D1 | — | 8.50 | — | mm | — | 0.335 | — | inches |
| E1 | — | 8.50 | — | mm | — | 0.335 | — | inches |
| e | — | 0.50 | — | mm | — | 0.020 | — | inches |
| b | 0.25 | 0.30 | 0.35 | mm | 0.010 | 0.012 | 0.014 | inches |
| aaa | | 0.10 | | mm | | 0.004 | | inches |
| bbb | | 0.10 | | mm | | 0.004 | | inches |
| ddd | | 0.08 | | mm | | 0.003 | | inches |
| eee | | 0.15 | | mm | | 0.006 | | inches |
| fff | | 0.05 | | mm | | 0.002 | | inches |
| MD/ME | | 18/18 | | mm | | 18/18 | | inches |

**Notes:**

1. Controlling dimension: Millimeters.
2. Primary DATUM C and seating plane are defined by the spherical crowns of the solder balls.
3. Dimension b is measured at the maximum solder ball diameter, parallel to primary DATUM C.
4. A minimum clearance of 0.25mm between the edge of the solder ball and the body edge is necessary.
5. Reference document: JEDEC MO–207.
6. The pattern of Pin 1 fiducial is for reference only.
7. Special characteristics C class: bbb, ccc.

# 9. Ordering Information

The order number AR6001X-BC1B specifies a current version of the AR6001X.

Document Number: 981-00032-001 MKG-0190 Rev. 1


**Atheros Communications, Incorporated**
5480 Great America Parkway
Santa Clara, CA 95054
t: 408/773-5200
f: 408/773-9940
www.atheros.com