

---

## Bluetooth Low Energy (BLE) System on Chip

---

### Key Features

- ARM® Cortex™-M0 32-bit processor
- Memory
  - 512/256KB in-system flash memory
  - 128KB ROM
  - 138KB SRAM, all programmable retention in sleep mode
  - 8-channel DMA
- 33/19 general purpose I/O pins
  - All pins can be configured as serial interface and programmable IO MUX function mapping
  - All pins can be configured for wake-up
  - 18 pins for triggering interrupt
  - 3 quadrature decoder(QDEC)
  - 6-channel PWM
  - 4-channel I2S
  - 2-channel PDM
  - 2-channel I2C
  - 2-channel SPI
  - 1-channel UART
  - JTAG
- 8-channel 12bit ADC with analog PGA
- 4-channel 32bit timer, one watchdog timer
- Real timer counter (RTC)
- Power, clock, reset controller
- Flexible power management
  - Supply voltage range 1.8V to 3.6V
  - Embedded buck DC-DC
  - Embedded LDOs
  - Battery monitor: Supports low battery detection
  - 2µA @ Sleep Mode with 32KHz RTC
  - 0.7µA @ OFF Mode(IO wake up only)
- 2.4 GHz transceiver
  - Compliant to Bluetooth 5.0, ETSI EN 300 328 and EN 300 440 Class 2 (Europe), FCC CFR47 Part 15 (US) and ARIB STD-T66 (Japan)
  - Sensitivity:
    - 97dBm@BLE 1Mbps data rate
    - 103dBm@BLE 125Kbps data rate
  - TX Power -20 to +10dBm in 3dB steps
  - Receiver: 8mA @sensitivity level
  - Transmitter: 8mA @0dBm TX power
  - Single-pin antenna: no RF matching or RX/TX switching required
  - RSSI (1dB resolution)
- RC oscillator hardware calibrations
  - 32KHz RC oscillator automatic calibration
  - 32MHz RC oscillator automatic calibration
- AES-128 encryption hardware
  - AES-ECB
  - AES-CCM
- Link layer hardware
  - Automatic packet assembly
  - Automatic packet detection and validation
  - Auto Re-transmit
  - Auto ACK
  - Hardware Address Matching
  - Random number generator
- Operating temperature: -40 °C~125°C
- RoHS Package: QFN48/ QFN32
- Applications: wearables, beacons, appliances, home and building, health and medical, sports and fitness, industrial and manufacturing, retail and payment, security, data transmission, remote control, PC/mobile/TV peripherals, internet of things (IoT)

## Table of Content

|                |  |           |
|----------------|--|-----------|
| <b>1</b>       | <b>Introduction .....</b>                          | <b>4</b>  |
| <b>2</b>       | <b>Product Overview .....</b>                      | <b>5</b>  |
| <b>2.1</b>     | <b>Block Diagram .....</b>                         | <b>5</b>  |
| <b>2.2</b>     | <b>Pin Assignments and Functions .....</b>         | <b>6</b>  |
| <b>2.2.1</b>   | <b>CMT4501 (QFN48) .....</b>                       | <b>6</b>  |
| <b>2.2.1.1</b> | <b>Pin Assignment .....</b>                        | <b>6</b>  |
| <b>2.2.1.2</b> | <b>Pin Function .....</b>                          | <b>6</b>  |
| <b>2.2.2</b>   | <b>CMT4501 (QFN32) .....</b>                       | <b>9</b>  |
| <b>2.2.2.1</b> | <b>Pin Assignment .....</b>                        | <b>9</b>  |
| <b>2.2.2.2</b> | <b>Pin Functions.....</b>                          | <b>9</b>  |
| <b>3</b>       | <b>System Blocks .....</b>                         | <b>10</b> |
| <b>3.1</b>     | <b>CPU .....</b>                                   | <b>10</b> |
| <b>3.2</b>     | <b>Memory .....</b>                                | <b>11</b> |
| <b>3.2.1</b>   | <b>ROM Memory .....</b>                            | <b>13</b> |
| <b>3.2.2</b>   | <b>SRAM Memory .....</b>                           | <b>13</b> |
| <b>3.2.3</b>   | <b>Memory Address Mapping .....</b>                | <b>13</b> |
| <b>3.2.4</b>   | <b>FLASH Memory.....</b>                           | <b>13</b> |
| <b>3.3</b>     | <b>Boot and Execution Modes.....</b>               | <b>13</b> |
| <b>3.3.1</b>   | <b>Mirror Mode .....</b>                           | <b>14</b> |
| <b>3.3.2</b>   | <b>FLASH Mode.....</b>                             | <b>14</b> |
| <b>3.3.3</b>   | <b>Boot loader .....</b>                           | <b>14</b> |
| <b>3.4</b>     | <b>Power, Clock and Reset (PCR) .....</b>          | <b>15</b> |
| <b>3.5</b>     | <b>Power Management (POWER).....</b>               | <b>15</b> |
| <b>3.6</b>     | <b>Low Power Features .....</b>                    | <b>17</b> |
| <b>3.6.1</b>   | <b>Operation and Sleep States .....</b>            | <b>17</b> |
| <b>3.6.1.1</b> | <b>Normal State .....</b>                          | <b>17</b> |
| <b>3.6.1.2</b> | <b>Clock Gate State .....</b>                      | <b>17</b> |
| <b>3.6.1.3</b> | <b>System Sleep State .....</b>                    | <b>17</b> |
| <b>3.6.1.4</b> | <b>System Off State .....</b>                      | <b>17</b> |
| <b>3.6.2</b>   | <b>State Transition .....</b>                      | <b>17</b> |
| <b>3.6.2.1</b> | <b>Entering Clock Gate State and Wake-up .....</b> | <b>17</b> |
| <b>3.6.2.2</b> | <b>Entering Sleep/off States and Wake-up .....</b> | <b>17</b> |
| <b>3.7</b>     | <b>Interrupts.....</b>                             | <b>17</b> |

|               |   |           |
|---------------|---|-----------|
| <b>3.8</b>    | <b>Clock Management (CLOCK)</b> .....   | <b>18</b> |
| <b>3.9</b>    | <b>IOMUX</b> .....  | <b>20</b> |
| <b>3.10</b>   | <b>GPIO</b> .....   | <b>31</b> |
| <b>4</b>      | <b>Peripheral Blocks</b> .....  | <b>35</b> |
| <b>4.1</b>    | <b>2.4GHz Radio</b> .....   | <b>35</b> |
| <b>4.2</b>    | <b>Timer/Counters (TIMER)</b> .....   | <b>35</b> |
| <b>4.3</b>    | <b>Real Time Counter (RTC)</b> .....  | <b>38</b> |
| <b>4.4</b>    | <b>AES-ECB Encryption (ECB)</b> .....   | <b>40</b> |
| <b>4.5</b>    | <b>Random Number Generator (RNG)</b> .....  | <b>42</b> |
| <b>4.6</b>    | <b>Watchdog Timer (WDT)</b> .....   | <b>42</b> |
| <b>4.7</b>    | <b>SPI (SPI)</b> .....  | <b>42</b> |
| <b>4.8</b>    | <b>I2C (I2c0, I2c1 Two Independent Instances)</b> .....                               | <b>47</b> |
| <b>4.9</b>    | <b>I2S</b> .....  | <b>56</b> |
| <b>4.10</b>   | <b>UART (UART)</b> .....  | <b>62</b> |
| <b>4.11</b>   | <b>Pulse Width Modulation (PWM)</b> .....   | <b>72</b> |
| <b>4.12</b>   | <b>Quadrature Decoder (QDEC)</b> .....  | <b>80</b> |
| <b>4.13</b>   | <b>Key Scan (KSCAN)</b> .....   | <b>84</b> |
| <b>4.14</b>   | <b>Analog to Digital Converter (ADC) with Programmable Gain Amplifier (PGA)</b> ..... | <b>85</b> |
| <b>4.14.1</b> | <b>PGA Path</b> .....   | <b>86</b> |
| <b>4.14.2</b> | <b>ADC Path</b> .....   | <b>86</b> |
| <b>4.14.3</b> | <b>ADC Channel &lt;3:0&gt; Connectivity</b> .....                                     | <b>89</b> |
| <b>5</b>      | <b>Absolute Maximum Ratings</b> .....   | <b>90</b> |
| <b>6</b>      | <b>Operating Conditions</b> .....   | <b>92</b> |
| <b>7</b>      | <b>Radio Transceiver</b> .....  | <b>92</b> |
| <b>7.1</b>    | <b>Radio Current Consumption</b> .....  | <b>92</b> |
| <b>7.2</b>    | <b>Transmitter Specification</b> .....  | <b>92</b> |
| <b>7.3</b>    | <b>Receiver Specification</b> .....   | <b>92</b> |
| <b>7.3.1</b>  | <b>RX BLE 1Mbps GFSK</b> .....  | <b>93</b> |
| <b>7.3.2</b>  | <b>RX BLE 2Mbps GFSK</b> .....  | <b>93</b> |
| <b>7.3.3</b>  | <b>RX 500Kbps GFSK</b> .....  | <b>94</b> |
| <b>7.3.4</b>  | <b>RX 125Kbps GFSK</b> .....  | <b>94</b> |
| <b>7.4</b>    | <b>RSSI Specifications</b> .....  | <b>95</b> |
| <b>8</b>      | <b>Glossary</b> .....   | <b>96</b> |
| <b>9</b>      | <b>Package dimensions</b> .....   | <b>97</b> |
| <b>10</b>     | <b>Sample Application and Layout Guide</b> .....                                      | <b>98</b> |

# **1 Introduction**

CMT4501 is a System on Chip (SoC) for Bluetooth® low energy applications. CMT4501 has 32-bit ARM® Cortex™-M0 CPU with 138KSRAM/Retention SRAM and an ultra-low power, high performance, multi-mode radio. CMT4501 can support BLE with security, application and over-the-air download update. Serial peripheral IO and integrated application IP enables customer product to be built with minimum bill-of-material (BOM) cost.

## 2 Product Overview

### 2.1 Block Diagram

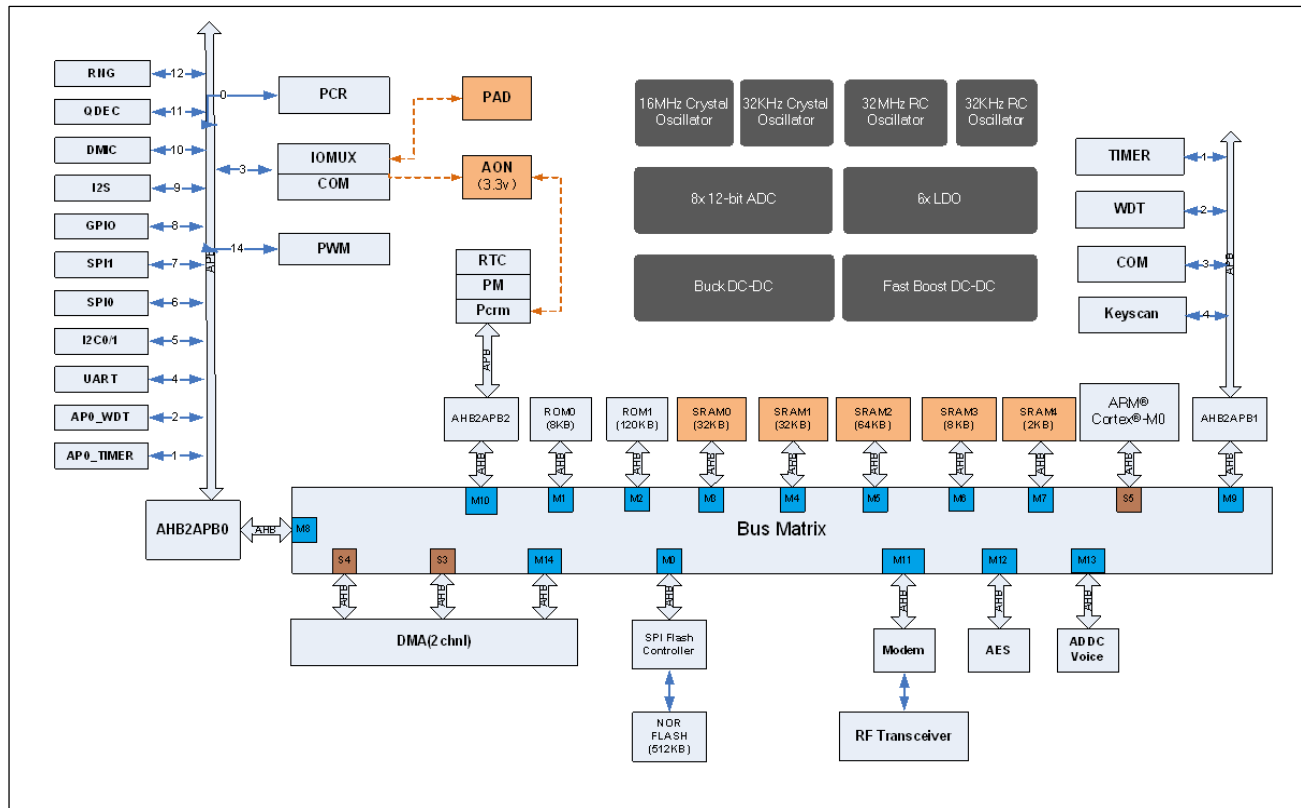


Figure 1: CMT4501 block diagram

## 2.2 Pin Assignments and Functions

This section describes the pin assignment and the pin functions for the different package types.

### 2.2.1 CMT4501 (QFN48)

#### 2.2.1.1 Pin Assignment

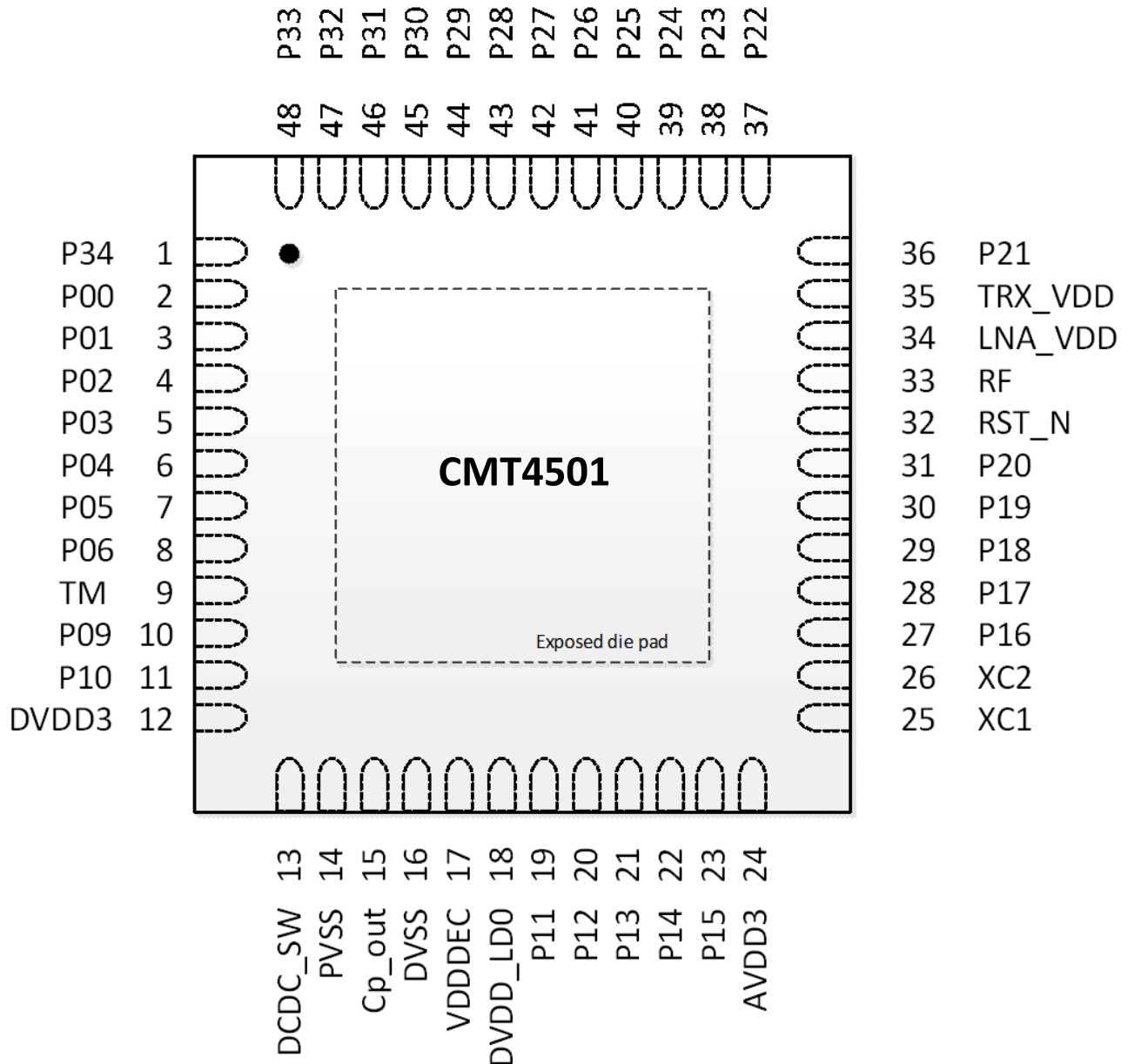


Figure 2: Pin assignment - CMT4501 QFN48 package

#### 2.2.1.2 Pin Function

| Pin | Pin name | Description  |
|-----|----------|--|
| 1   | P34      | all functions configurable<br>*Note: Not support interrupt and ADC function                                |
| 2   | P00      | all functions configurable/ JTAG_TDO<br>*Note: Not support ADC function                                    |
| 3   | P01      | all functions configurable/ JTAG_TDI<br>*Note: Not support ADC function                                    |
| 4   | P02      | all functions configurable/JTAG_TMS<br>*Note: Not support ADC function                                     |
| 5   | P03      | all functions configurable/JTAG_TCK<br>*Note: Not support ADC function                                     |
| 6   | P04      | all functions configurable<br>*Note: Not support ADC function  |
| 7   | P05      | all functions configurable<br>*Note: Not support ADC function  |
| 8   | P06      | all functions configurable<br>*Note: Not support ADC function  |
| 9   | TM       | Test_Mode  |
| 10  | P09      | all functions configurable<br>*Note: Not support ADC function  |
| 11  | P10      | all functions configurable<br>*Note: Not support ADC function  |
| 12  | DVDD3    | 3V power supply for digital IO, DCDC, Charge pump  |
| 13  | DCDC_SW  | Buck dcdc output   |
| 14  | PVSS     | Buck dcdc and charge pump power vss  |
| 15  | cp_out   | charge pump output   |
| 16  | DVSS     | digital vss  |
| 17  | VDDDEC   | 1.2V VDD_CORE, digital LDO output  |
| 18  | DVDD_LDO | digital LDO input  |
| 19  | P11      | all functions configurable/AIO<0>  |
| 20  | P12      | all functions configurable/AIO<1>  |
| 21  | P13      | all functions configurable/AIO<2>  |
| 22  | P14      | all functions configurable/AIO<3>  |
| 23  | P15      | all functions configurable/AIO<4>  |
| 24  | AVDD3    | 3V power supply for analog IO, bg, rcosc, etc  |
| 25  | XC1      | 16M crystal input  |
| 26  | XC2      | 16M crystal output   |
| 27  | P16      | all functions configurable/AIO<5>/32K crystal input  |
| 28  | P17      | all functions configurable/AIO<6>/32k crystal output   |
| 29  | P18      | all functions configurable/AIO<7>/PGA differential positive input<br>*Note: Not support interrupt function |
| 30  | P19      | all functions configurable/AIO<8>/PGA differential negative input<br>*Note: Not support interrupt function |
| 31  | P20      | all functions configurable/AIO<9>/Micphone bias output<br>*Note: Not support interrupt function            |

|    |         |  |
|----|---------|--|
| 32 | RST_N   | reset pin  |
| 33 | RF      | RF antenna   |
| 34 | LNA_VDD | LNA_VDD  |
| 35 | TRX_VDD | TRX_VDD  |
| 36 | P21     | all functions configurable<br>*Note: Not support interrupt function and ADC function                     |
| 37 | P22     | all functions configurable<br>*Note: Not support interrupt function and ADC function                     |
| 38 | P23     | all functions configurable<br>*Note: Not support interrupt function and ADC function                     |
| 39 | P24     | all functions configurable/test_mode_select[0]<br>*Note: Not support interrupt function and ADC function |
| 40 | P25     | all functions configurable/test_mode_select[1]<br>*Note: Not support interrupt function and ADC function |
| 41 | P26     | all functions configurable<br>*Note: Not support interrupt function and ADC function                     |
| 42 | P27     | all functions configurable<br>*Note: Not support interrupt function and ADC function                     |
| 43 | P28     | all functions configurable<br>*Note: Not support interrupt function and ADC function                     |
| 44 | P29     | all functions configurable<br>*Note: Not support interrupt function and ADC function                     |
| 45 | P30     | all functions configurable<br>*Note: Not support interrupt function and ADC function                     |
| 46 | P31     | all functions configurable<br>*Note: Not support interrupt function and ADC function                     |
| 47 | P32     | all functions configurable<br>*Note: Not support interrupt function and ADC function                     |
| 48 | P33     | all functions configurable<br>*Note: Not support interrupt function and ADC function                     |

**Table 1: Pin functions CMT4501 QFN48 package**



## 2.2.2 CMT4501 (QFN32)

### 2.2.2.1 Pin Assignment

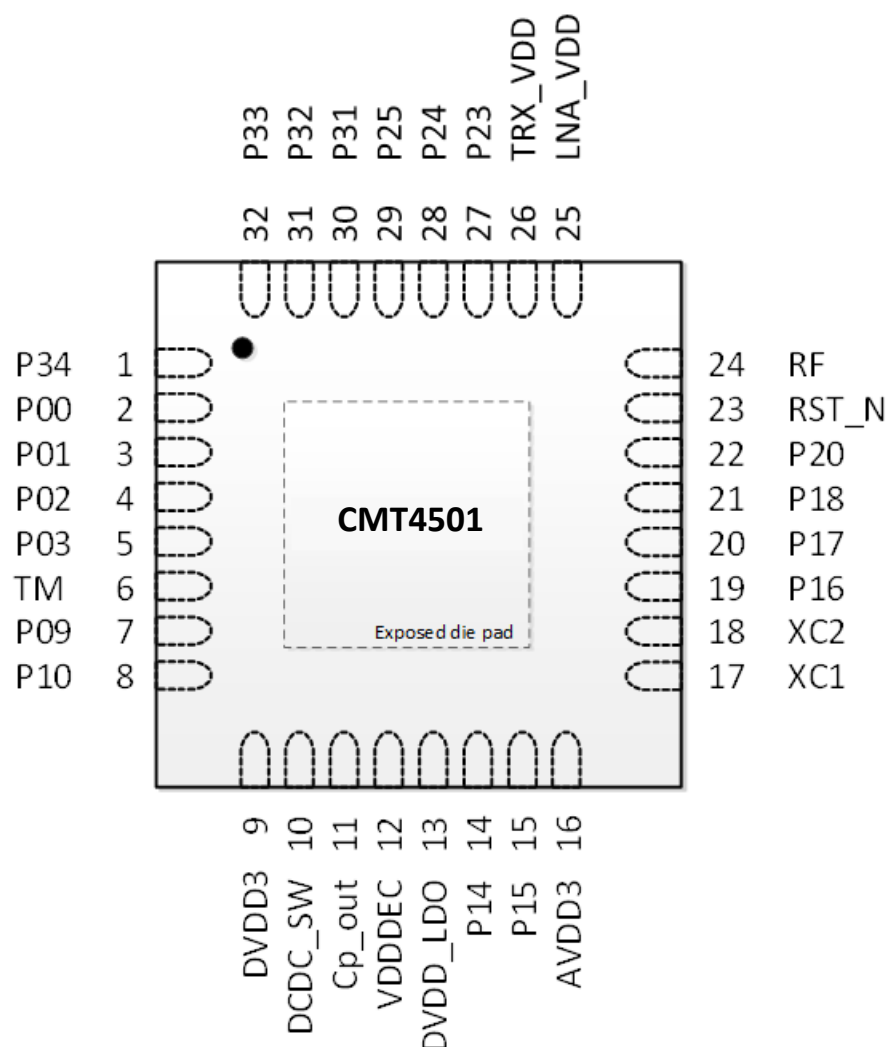


Figure 3: Pin assignment - CMT4501 QFN32 package

### 2.2.2.2 Pin Functions

| Pin | Pin name | Description   |
|-----|----------|---|
| 1   | P34      | all functions configurable<br>*Note: Not support interrupt and ADC function |
| 2   | P00      | all functions configurable/ JTAG_TDO<br>*Note: Not support ADC function     |
| 3   | P01      | all functions configurable/ JTAG_TDO<br>*Note: Not support ADC function     |
| 4   | P02      | all functions configurable/ JTAG_TDO<br>*Note: Not support ADC function     |
| 5   | P03      | all functions configurable/ JTAG_TDO<br>*Note: Not support ADC function     |
| 6   | TM       | Test_Mode   |
| 7   | P09      | all functions configurable/ JTAG_TDO<br>*Note: Not support ADC function     |

|    |          |   |
|----|----------|---|
| 8  | P10      | all functions configurable/ JTAG_TDO<br>*Note: Not support ADC function                         |
| 9  | DVDD3    | 3V power supply for digital IO, DCDC, Charge pump   |
| 10 | DCDC_SW  | Buck dcdc output  |
| 11 | cp_out   | charge pump output  |
| 12 | VDDDEC   | 1.2V VDD_CORE, digital LDO output   |
| 13 | DVDD_LDO | digital LDO input   |
|    |          |   |
| 14 | P14      | all functions configurable/AIO<3>   |
| 15 | P15      | all functions configurable/AIO<4>   |
| 16 | AVDD3    | 3V power supply for analog IO, bg, rcosc, etc   |
| 17 | XC1      | 16M crystal input   |
| 18 | XC2      | 16M crystal output  |
| 19 | P16      | all functions configurable/AIO<5>/32K crystal input   |
| 20 | P17      | all functions configurable/AIO<6>/32k crystal output  |
| 21 | P18      | all functions configurable/ JTAG_TDO<br>*Note: Not support interrupt function                   |
| 22 | P20      | all functions configurable/AIO<9>/Micphone bias output  |
| 23 | RST_N    | reset pin   |
| 24 | RF       | RF antenna  |
| 25 | LNA_VDD  | LNA_VDD   |
| 26 | TRX_VDD  | TRX_VDD   |
| 27 | P23      | all functions configurable<br>*Note: Not support interrupt and ADC function                     |
| 28 | P24      | all functions configurable/test_mode_select[0]<br>*Note: Not support interrupt and ADC function |
| 29 | P25      | all functions configurable/test_mode_select[1]<br>*Note: Not support interrupt and ADC function |
| 30 | P31      | all functions configurable<br>*Note: Not support interrupt and ADC function                     |
| 31 | P32      | all functions configurable<br>*Note: Not support interrupt and ADC function                     |
| 32 | P33      | all functions configurable<br>*Note: Not support interrupt and ADC function                     |

**Table 2: Pin functions CMT4501 QFN32 package**

## 3 System Blocks

The system block diagram of CMT4501 is shown in **Figure 1**.

### 3.1 CPU

The CMT4501 has an ARM Cortex-M0 CPU. The CPU, memories, and all peripherals are connected by AMBA bus fabrics.

The ARM® Cortex™-M0 CPU has a 16-bit instruction set with 32-bit extensions (Thumb-2® technology) that delivers high-density code with a small-memory-footprint. By using a single-cycle 32-bit multiplier, a 3-stage pipeline and a Nested Vector Interrupt Controller (NVIC), the ARM Cortex™-M0 CPU makes program execution simple and highly efficient.

The CPU will play controller role in BLE modem and run all user applications. The following main features

are listed below.

- Up to 48Mhz ARM Cortex™-M0 processor core.
  - Low gate count and high energy efficient.
  - ARMv6M architecture, Thumb ISA but no ARM ISA.
  - No cache and no TCM.
  - Up to 32 interrupts embedded NVIC.
  - SysTick timer.
  - Sleep/deep sleep mode.
  - Support low power WFI and WFE
- 4 32-bit general purpose timers and 1 watchdog timer (WDT).
- 120KB ROM for boot and protocol stack.
- 138KB retention SRAM for program and data.
- AHB to APB Bridge for peripherals and registers.
- Clock and reset controller.
- AHB debug access port interface and DAP ROM.
- APB interface to/from BLE modem.
- Dynamic and static clock gating to save power.
- No TRACE.

Some of these features are shared with the AP subsystem.

## **3.2 Memory**

CMT4501 has total 128KB ROM, 138KB SRAM and up to 512KB FLASH. The physical address space of these memories is shown in Figure4.

## CMT4501 Memory Space

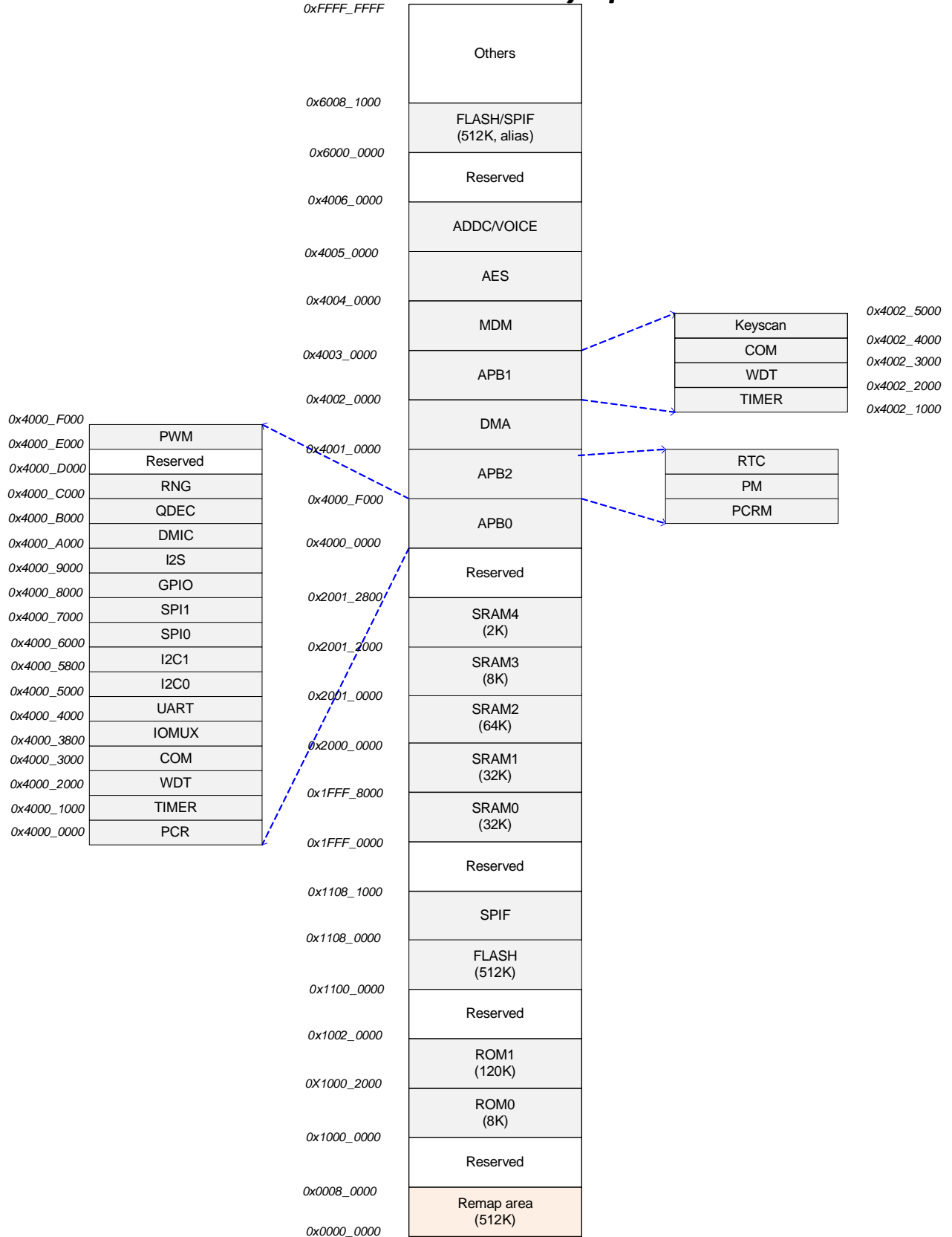


Figure 4: CMT4501 memory space

### 3.2.1 ROM Memory

CMT4501 has 2 ROMs.

|      | SIZE  | CONTENT   |
|------|-------|---|
| ROM0 | 8KB   | Reserved  |
| ROM1 | 120KB | Boot ROM for M0.<br>Protocol stack.<br>Common peripheral drivers. |

Table 3: List of ROMs

### 3.2.2 SRAM Memory

CMT4501 has 5 SRAM blocks. All 5 SRAM blocks have retention capability, which can be configured individually. All SRAM blocks can be used to store program or data.

|       | SIZE | CONTENT |
|-------|------|---------|
| SRAM0 | 32KB |         |
| SRAM1 | 32KB |         |
| SRAM2 | 64KB |         |
| SRAM3 | 8KB  |         |
| SRAM4 | 2KB  |         |

Table 4: List of SRAMs

### 3.2.3 Memory Address Mapping

| Name  | Size (KB) | Master | Physical Address    | CM4 Alias | M0 Remap |     |     |
|-------|-----------|--------|---------------------|-----------|----------|-----|-----|
|       |           |        |                     |           | 0        | 1   | 2   |
| ROM0  | 8         | M0     | 1000_0000~1000_1FFF | 0x0       |          |     |     |
| ROM1  | 120       | M0     | 1000_2000~1001_FFFF |           | 0x0      |     |     |
| RAM0  | 32        | M0     | 1FFF_0000~1FFF_7FFF |           |          |     |     |
| RAM1  | 32        | M0     | 1FFF_8000~1FFF_FFFF |           |          |     |     |
| RAM2  | 64        | M0     | 2000_0000~2000_FFFF |           |          | 0x0 |     |
| RAM3  | 8         | M0     | 2001_0000~2001_1FFF |           |          |     |     |
| RAM4  | 2         | M0     | 2001_2000~2001_27FF |           |          |     |     |
| FLASH | 512       | M0     | 1100_0000~1107_FFFF |           |          |     | 0x0 |
|       |           |        | 6000_0000~6007_FFFF |           |          |     |     |

Table 5: Memory address mapping

### 3.2.4 FLASH Memory

CMT4501 has FLASH to provide non-volatile program and data storage. The size of the FLASH can be 256KB or 512KB. CMT4501 supports 2-wire reading.

## 3.3 Boot and Execution Modes

During the boot, the ROM1 is aliased to 0x0 address. The M0 starts to execute the program from the ROM1.

# Boot

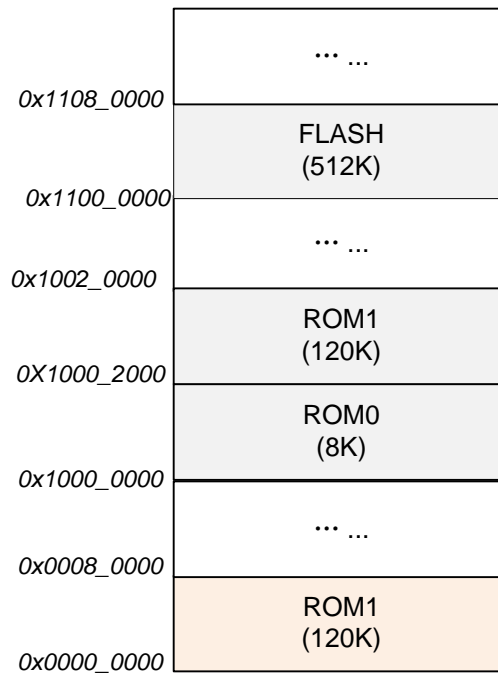


Figure 5: CMT4501 boot mode

## 3.3.1 Mirror Mode

The mirror mode is not tied to the chip variations. Any chip variation can use mirror mode to execute program. In the mirror mode, the program is copied from the FLASH to the SRAM, then is executed in the SRAM. For the M0 processor, one of the SRAM blocks must be aliased to 0x0 address.

## 3.3.2 FLASH Mode

The FLASH mode is not tied to the chip variations. Any chip variation can use FLASH mode to execute program. In the FLASH mode, the program is executed in the FLASH. For the M0 processor, the FLASH must be aliased to 0x0 address.

## 3.3.3 Boot loader

The boot loader in the ROM has the basic structure as shown below. The content in the FLASH should be specifically defined to allow boot loader to identify whether the FLASH content is valid, as shown in the example below. If the FLASH is valid, the ROM boot loader will put the chip in the normal mode and start normal program execution. If the FLASH is not valid, the boot loader will enter FLASH programming mode.

| Address | Variable     | Content                       |
|---------|--------------|-------------------------------|
| 0       | PRODUCT_MODE | Identify the chip mode        |
| 4       | CODE_BASE    | The base address of the code  |
| 8       | CODE_LEN     | The length of the code        |
| C       | BOOT_MODE    | Identify mirror or FLASH mode |

Table 6: Flash content example

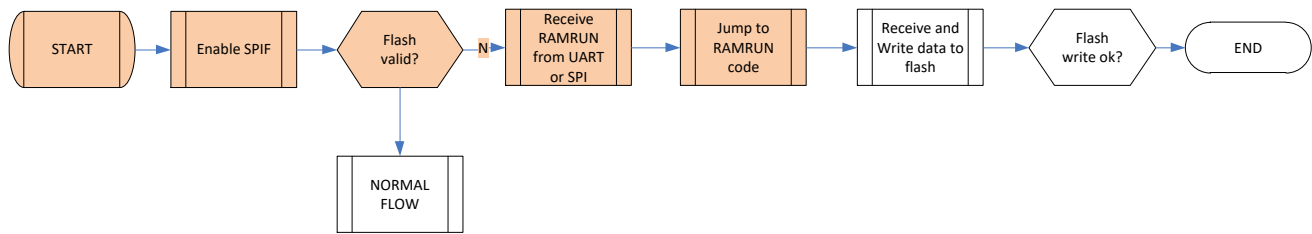


Figure6: Bootloader flow

### 3.4 Power, Clock and Reset (PCR)

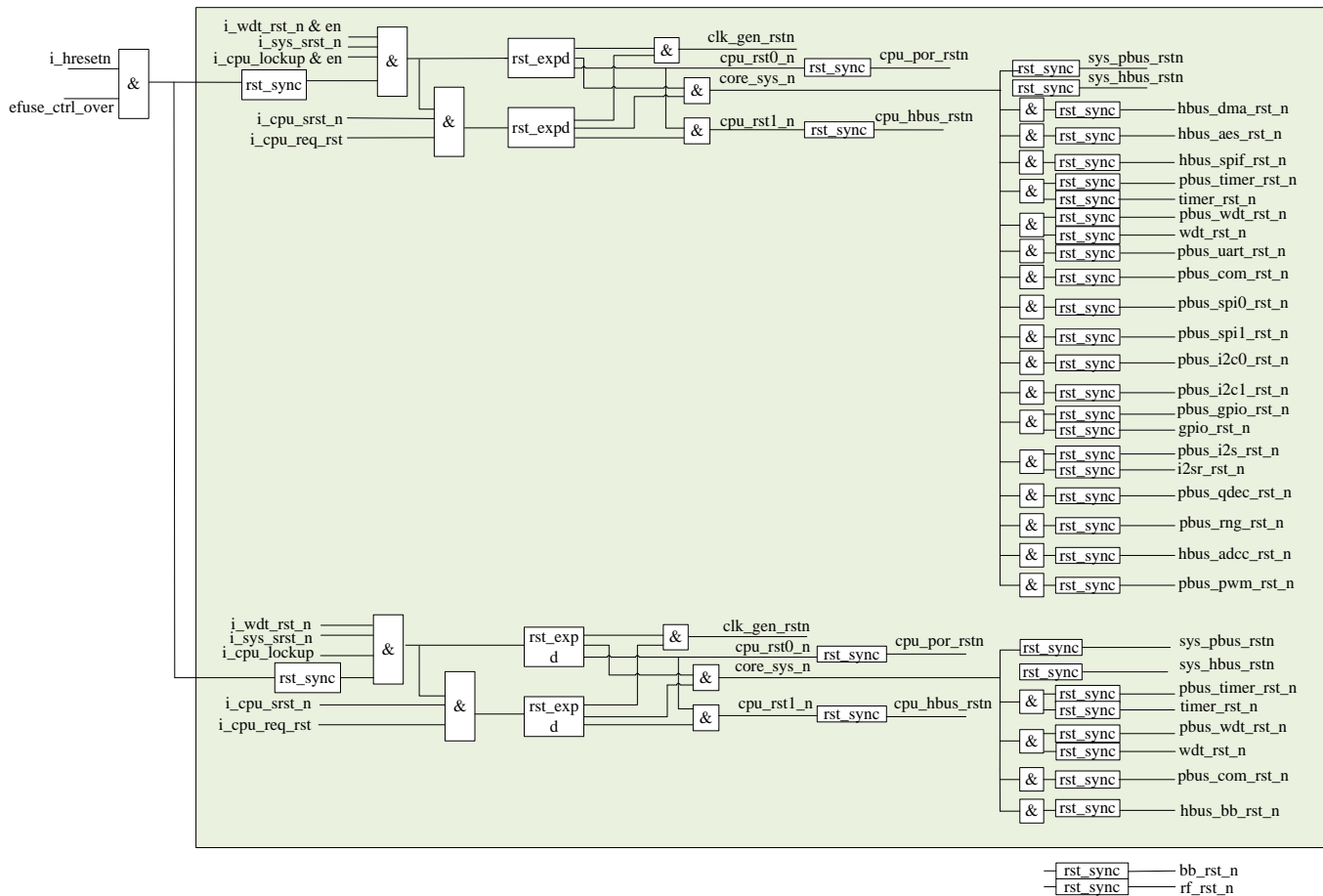


Figure 7: CMT4501 power, clock and reset

### 3.5 Power Management (POWER)

The power management system is highly flexible with functional blocks such as the CPU, radio transceiver, and peripherals saving separate power state control in addition to the System Sleep mode and OFF modes. When in System Normal mode, all functional blocks will independently be turned on depending on needed application functionality.

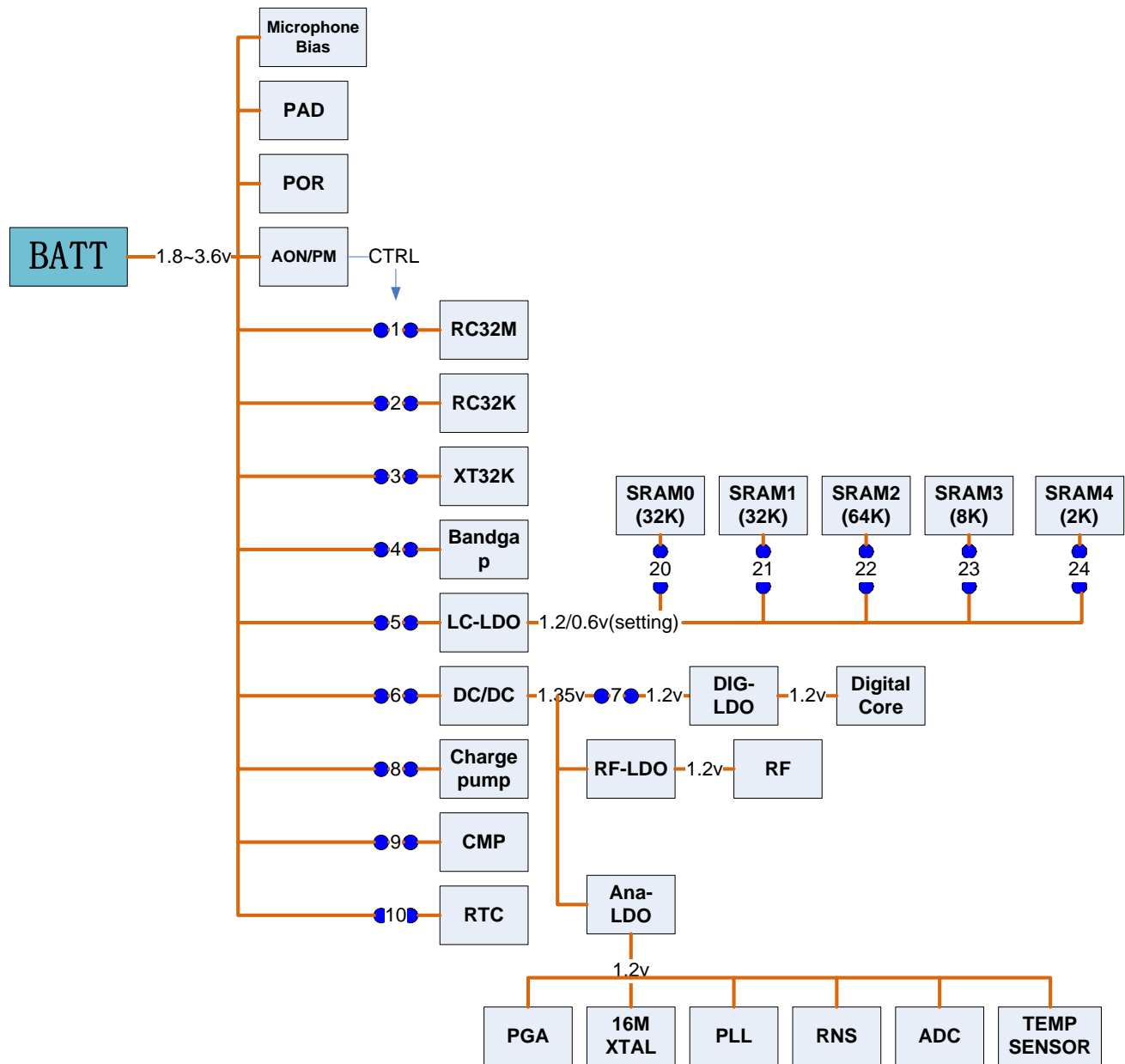


Figure 8: Power system

The following diagram is Normal, Sleep and Off mode. Switches are optional depending on user's request.

| Switch       | Normal | Sleep    | Off |
|--------------|--------|----------|-----|
| 1RC32M       | On     | Off      | Off |
| 2RC32K       | On     | Optional | Off |
| 3XT32K       | On     | Optional | Off |
| 4bandgap     | On     | Off      | Off |
| 5LC-LDO      | On     | on       | Off |
| 6DC/DC       | On     | Off      | Off |
| 7DIG-LDO     | On     | Off      | Off |
| 8charge pump | On     | Off      | Off |
| 9CMP         | On     | Optional | Off |



|            |      |          |     |
|------------|------|----------|-----|
| 10RTC      | On   | Optional | Off |
| 20SRAM-32K | 1.2v | 0.6v     | 0   |
| 21SRAM-32K | 1.2v | 0.6v     | 0   |
| 22SRAM-64K | 1.2v | 0.6v     | 0   |
| 23SRAM-8K  | 1.2v | 0.6v     | 0   |
| 24SRAM-2K  | 1.2v | 0.6v     | 0   |

Table 7: Flash Switches of different power modes

## 3.6 Low Power Features

### 3.6.1 Operation and Sleep States

#### 3.6.1.1 Normal State

#### 3.6.1.2 Clock Gate State

The CPU executes WFI/WFE to enter clock gate state. After wake-up from clock-gate state, the CPU continues to execute the program from where it stopped. The wake-up sources includes interrupts and events. The wake-up sources are configured by the software according to applications.

#### 3.6.1.3 System Sleep State

The wake-up sources include:

- IO
- RTC
- RESET
- UVLO reset

#### 3.6.1.4 System Off State

The wake-up sources include:

- IOs
- RESET
- UVLO reset

### 3.6.2 State Transition

#### 3.6.2.1 Entering Clock Gate State and Wake-up

CPU executes WFI/WFE.

#### 3.6.2.2 Entering Sleep/off States and Wake-up

The PM registers identify whether the CPU is in mirror mode or FLASH mode before sleep or off, and record the remap and vectors. The CPU configures the corresponding PM registers to put the chip into sleep or off mode. After wake-up, the chip enters boot mode to execute boot code in the ROM. The ROM code checks the mode before sleep/off and the remap information, perform corresponding configurations, and starts to execute the program.

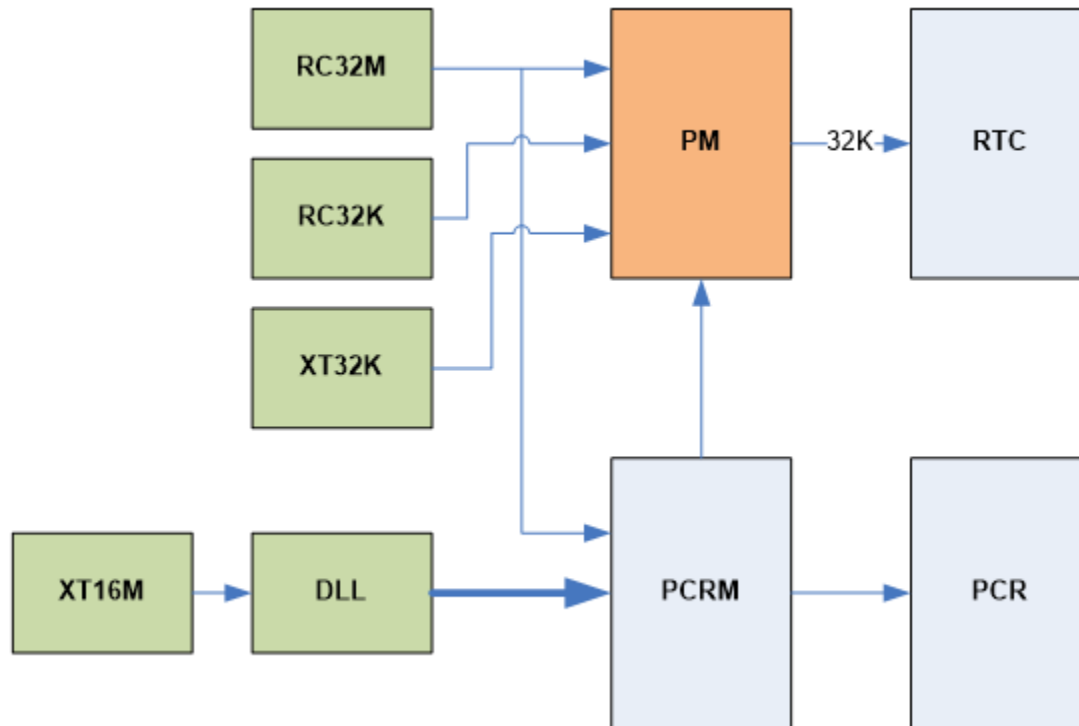
## 3.7 Interrupts

| Interrupt Name | M0 Interrupt Number |
|----------------|---------------------|
| Reserved       | 0                   |
| Reserved       | 1                   |
| cp_timer_irq   | 2                   |
| cp_wdt_irq     | 3                   |

|             |    |
|-------------|----|
| bb_irq      | 4  |
| kscan_irq   | 5  |
| rtc_irq     | 6  |
| Reserved    | 7  |
| Reserved    | 8  |
| timer_irq   | 9  |
| wdt_irq     | 10 |
| uart_irq    | 11 |
| i2c0_irq    | 12 |
| i2c1_irq    | 13 |
| spi0_irq    | 14 |
| spi1_irq    | 15 |
| gpio_irq    | 16 |
| i2s_irq     | 17 |
| spif_irq    | 18 |
| dmac_intr   | 19 |
| dmac_inttc  | 20 |
| dmac_interr | 21 |
| fpidc       | 22 |
| fpdzc       | 23 |
| fpioc       | 24 |
| fpufc       | 25 |
| fpofc       | 26 |
| fpixc       | 27 |
| aes_irq     | 28 |
| adcc_irq    | 29 |
| qdec_irq    | 30 |
| rng_irq     | 31 |

Table 8: Interrupts

### 3.8 Clock Management (CLOCK)



**Figure 10: Clock management**

There are two crystal clock sources: 16MHz crystal oscillator (XT16M) and 32.768kHz crystal oscillator (XT32k), of which the 32.768k crystal oscillator is optional. There are also two on chip RC oscillators: 32MHz RC oscillator (RC32M) and 32kHz RC oscillator (RC32k), both of which can be calibrated with respect to 16MHz crystal oscillator. If 32.768kHz crystal is not installed, RC32k oscillator would be periodically calibrated and used for RTC. At initial power up or wake up before XT16M oscillator starts up, RC32M is used as the main clock. An on-chip DLL generates higher frequency clocks such as 32/48/64/96MHz from the XT16M clock source.

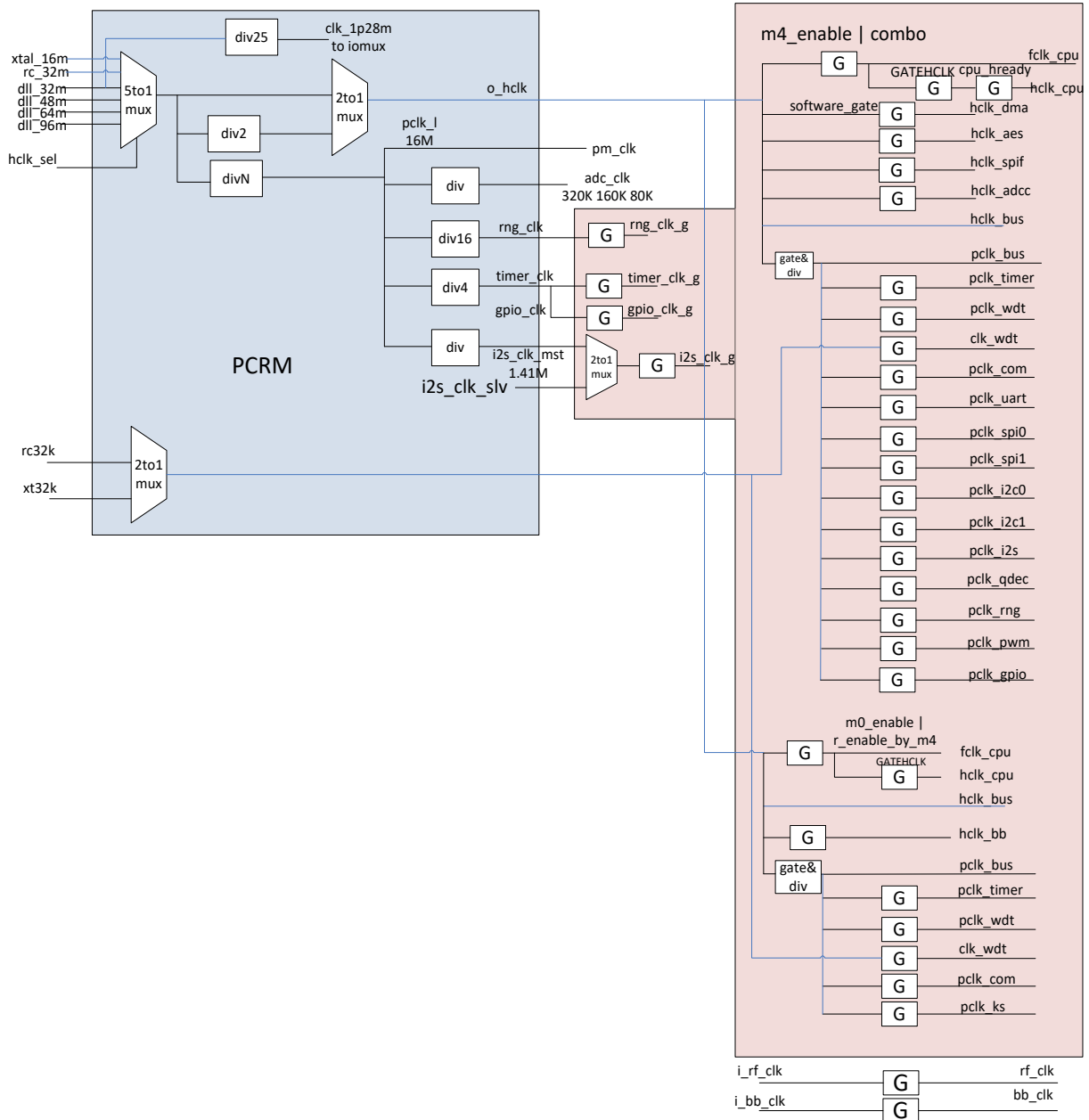
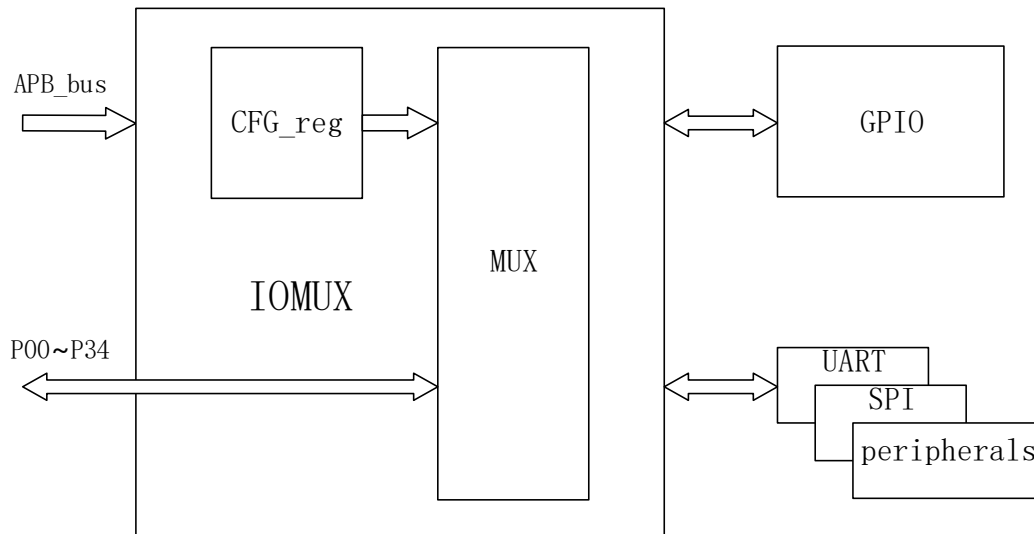


Figure11: Clock structure diagram

### 3.9 IOMUX

The IOMUX provides a flexible I/O configuration, as the ports of most of the peripherals can be configured and mapped to any of the physical I/O pads (I/O at die boundary). These peripheral modules include I2C 0-1, I2S, UART, PWM 0-5, SPI 0-1, Quadrature Decoder etc. However for other specific purpose peripherals, their I/Os mappings are fixed when they are enabled. These specific purpose peripherals include JTAG, analog\_ios, GPIOs and key scan.

Figure 12 below shows the IOMUX functional diagram.



**Figure12: IOMUX structure diagram**

There are 34 configurable pads which are from P00 to P07 and from P09 to P34. P08 pad is assigned for TM pin which is a test mode pin. The table below shows the mapping of the peripheral IOs that can be mapped through IOMUX. These include I2C 0-1, I2S, UART, PWM 0-5, SPI 0-1, Quadrature Decoder, 1.28MHz clock and dmico\_out.

| signal name | io | FULLMUX |
|-------------|----|---------|
| iic0_scl    | B  | 0       |
| iic0_sda    | B  | 1       |
| iic1_scl    | B  | 2       |
| iic1_sda    | B  | 3       |
| i2s_sck     | B  | 4       |
| i2s_ws      | B  | 5       |
| i2s_sdo0    | O  | 6       |
| i2s_sdo1    | O  | 35      |
| i2s_sdo2    | O  | 36      |
| i2s_sdo3    | O  | 37      |
| i2s_sdi0    | I  | 7       |
| i2s_sdi1    | I  | 38      |
| i2s_sdi2    | I  | 39      |
| i2s_sdi3    | I  | 40      |
| uart_tx     | O  | 8       |
| uart_rx     | I  | 9       |
| pwm0        | O  | 10      |
| pwm1        | O  | 11      |
| pwm2        | O  | 12      |
| pwm3        | O  | 13      |
| pwm4        | O  | 14      |

|               |   |    |
|---------------|---|----|
| pwm5          | O | 15 |
| spi_0_sck     | B | 16 |
| spi_0_ssn     | B | 17 |
| spi_0_tx      | O | 18 |
| spi_0_rx      | I | 19 |
| spi_1_sck     | B | 20 |
| spi_1_ssn     | B | 21 |
| spi_1_tx      | O | 22 |
| spi_1_rx      | I | 23 |
| chax          | I | 24 |
| chbx          | I | 25 |
| chix          | I | 26 |
| chay          | I | 27 |
| chby          | I | 28 |
| chiy          | I | 29 |
| chaz          | I | 30 |
| chbz          | I | 31 |
| chiz          | I | 32 |
| clk_1p28m     | O | 33 |
| adcc_dmic_out | I | 34 |

On the other hand, there are also special purpose peripherals, whose IOs are fixed to certain physical pads, when these peripheral functions are enabled. These special purpose peripherals include: JTAG, analog I/Os (ADC inputs), GPIO, and key scan. When they are enabled, their IOs are mapped to physical pads according to the following table (by default JTAG is enabled).

| QFN32 |    | QFN48     |            |      |              |            |
|-------|----|-----------|------------|------|--------------|------------|
| ✓     | 0  | GPIO_P00  | jtag_dout  | GPIO |              | mk_in[0]   |
| ✓     | 1  | GPIO_P01  | jtag_din   | GPIO |              | mk_out[0]  |
| ✓     | 2  | GPIO_P02  | jtag_tm    | GPIO |              | mk_in[1]   |
| ✓     | 3  | GPIO_P03  | jtag_clk   | GPIO |              | mk_out[1]  |
|       | 4  | GPIO_P04  | GPIO       |      |              | mk_out[9]  |
|       | 5  | GPIO_P05  | GPIO       |      |              | mk_in[10]  |
|       | 6  | GPIO_P06  | GPIO       |      |              | mk_out[10] |
|       | 7  | GPIO_P07  | GPIO       |      |              | mk_in[11]  |
| ✓     | 8  | TEST_MODE |            |      |              |            |
| ✓     | 9  | GPIO_P09  | GPIO       |      |              | mk_out[4]  |
| ✓     | 10 | GPIO_P10  | GPIO       |      |              | mk_in[4]   |
|       | 11 | GPIO_P11  | GPIO       |      | analog_io[0] | mk_out[11] |
|       | 12 | GPIO_P12  | GPIO       |      | analog_io[1] | mk_in[12]  |
|       | 13 | GPIO_P13  | GPIO       |      | analog_io[2] | mk_out[12] |
| ✓     | 14 | GPIO_P14  | GPIO       |      | analog_io[3] | mk_out[2]  |
| ✓     | 15 | GPIO_P15  | GPIO       |      | analog_io[4] | mk_in[2]   |
| ✓     | 16 | GPIO_P16  | XTALI(ANA) | GPIO |              | mk_out[16] |
| ✓     | 17 | GPIO_P17  | XTALO(ANA) | GPIO |              | mk_out[17] |
| ✓     | 18 | GPIO_P18  | GPIO       |      | analog_io[7] | mk_in[5]   |
|       | 19 | GPIO_P19  | GPIO       |      | analog_io[8] | mk_in[13]  |

|   |    |          |           |      |              |            |
|---|----|----------|-----------|------|--------------|------------|
| ✓ | 20 | GPIO_P20 | GPIO      |      | analog_io[9] | mk_out[5]  |
|   | 21 | GPIO_P21 | GPIO      |      |              | mk_out[13] |
|   | 22 | GPIO_P22 | GPIO      |      |              | mk_in[14]  |
| ✓ | 23 | GPIO_P23 | GPIO      |      |              | mk_in[6]   |
| ✓ | 24 | GPIO_P24 | GPIO      |      |              | mk_out[3]  |
| ✓ | 25 | GPIO_P25 | GPIO      |      |              | mk_in[3]   |
|   | 26 | GPIO_P26 | GPIO      |      |              | mk_out[14] |
|   | 27 | GPIO_P27 | GPIO      |      |              | mk_in[9]   |
|   | 28 | GPIO_P28 | GPIO      |      |              | mk_out[8]  |
|   | 29 | GPIO_P29 | GPIO      |      |              | mk_in[15]  |
|   | 30 | GPIO_P30 | GPIO      |      |              | mk_out[15] |
| ✓ | 31 | GPIO_P31 | spi_t_ssn | GPIO |              | mk_out[7]  |
| ✓ | 32 | GPIO_P32 | spi_t_rx  | GPIO |              | mk_in[7]   |
| ✓ | 33 | GPIO_P33 | spi_t_tx  | GPIO |              | mk_out[6]  |
| ✓ | 34 | GPIO_P34 | spi_t_sck | GPIO |              | mk_in[8]   |

In the IO Mux table above, the first column is the IO pad mapping in default mode, when no IOMUX function is selected and no special purpose peripherals such as analogIO, GPIO<0:3>, key scan, are enabled. In this mode, pin<0:3> are used for JTAG.

When analog IOs are enabled, pins<11:15>, <18:20> are connected to internal analog IOs. More specifically, analog\_io<0:4><9> are connected to ADC inputs, analog\_io<7,8> are connected to PGA inputs.

In JTAG mode, data output for JTAG test mode is mapped to P00; data input for JTAG test mode is mapped to P01; mode control input for JTAG test mode is mapped to P02; clock input for JTAG test mode is mapped to P03.

Detailed IOMUX register table and physical IO pad control are shown below.

Base address: 4000\_3800

| OFFSET  | TYPE | RESET  | NAME                | DESCRIPTION                        |
|---------|------|--------|---------------------|------------------------------------|
| 0x0     |      |        | r_analog_io         |                                    |
| [31:10] | RW   | 22'h0  | reserved            |                                    |
| [9:0]   | RW   | 10'h60 | r_analog_io_en      | Analog IO enable                   |
| 0xc     |      |        | full_mux0           | register description               |
| [31:0]  | RW   | 32'h0  | r_func_io_en[31:0]  | full mux enable. [8] must set to 0 |
| 0x10    |      |        | full_mux1           | register description               |
| [31:3]  | RW   | 29'h0  | reserved            |                                    |
| [2:0]   | RW   | 3'h0   | r_func_io_en[34:32] | full mux enable                    |
| 0x14    |      |        | gpio_papb           | register description               |
| [31:17] | RW   | 15'h0  | reserved            |                                    |
| [16]    | RW   | 1'h0   | r_gpio_pb_16_en     | gpio_16 enable                     |
| [15]    | RW   | 1'h0   | r_gpio_pb_15_en     | gpio_15 enable                     |
| [14]    | RW   | 1'h0   | r_gpio_pb_14_en     | gpio_14 enable                     |

|        |    |      |                 |                |
|--------|----|------|-----------------|----------------|
| [13]   | RW | 1'h0 | r_gpio_pb_13_en | gpio_13 enable |
| [12:4] | RW | 9'h0 | reserved        |                |
| [3]    | RW | 1'h0 | r_gpio_pa_03_en | gpio_03 enable |
| [2]    | RW | 1'h0 | r_gpio_pa_02_en | gpio_02 enable |
| [1]    | RW | 1'h0 | r_gpio_pa_01_en | gpio_01 enable |
| [0]    | RW | 1'h0 | r_gpio_pa_00_en | gpio_00 enable |

| 0x18    |    |      | func_io0        | register description           |
|---------|----|------|-----------------|--------------------------------|
| [31:30] | RW | 2'h0 | reserved        |                                |
| [29:24] | RW | 6'h0 | r_func_io03_sel | pad 3 full mux function select |
| [23:22] | RW | 2'h0 | reserved        |                                |
| [21:16] | RW | 6'h0 | r_func_io02_sel | pad 2 full mux function select |
| [15:14] | RW | 2'h0 | reserved        |                                |
| [13:8]  | RW | 6'h0 | r_func_io01_sel | pad 1 full mux function select |
| [7:6]   | RW | 2'h0 | reserved        |                                |
| [5:0]   | RW | 6'h0 | r_func_io00_sel | pad 0 full mux function select |

| 0x1c    |    |      | func_io1        | register description           |
|---------|----|------|-----------------|--------------------------------|
| [31:30] | RW | 2'h0 | reserved        |                                |
| [29:24] | RW | 6'h0 | r_func_io07_sel | pad 7 full mux function select |
| [23:22] | RW | 2'h0 | reserved        |                                |
| [21:16] | RW | 6'h0 | r_func_io06_sel | pad 6 full mux function select |
| [15:14] | RW | 2'h0 | reserved        |                                |
| [13:8]  | RW | 6'h0 | r_func_io05_sel | pad 5 full mux function select |
| [7:6]   | RW | 2'h0 | reserved        |                                |
| [5:0]   | RW | 6'h0 | r_func_io04_sel | pad 4 full mux function select |

| 0x20    |    |      | func_io2        | register description                                 |
|---------|----|------|-----------------|--|
| [31:30] | RW | 2'h0 | reserved        |  |
| [29:24] | RW | 6'h0 | r_func_io11_sel | pad 11 full mux function select                      |
| [23:22] | RW | 2'h0 | reserved        |  |
| [21:16] | RW | 6'h0 | r_func_io10_sel | pad 10 full mux function select                      |
| [15:14] | RW | 2'h0 | reserved        |  |
| [13:8]  | RW | 6'h0 | r_func_io09_sel | pad 9 full mux function select                       |
| [7:6]   | RW | 2'h0 | reserved        |  |
| [5:0]   | RW | 6'h0 | r_func_io08_sel | pad 8 full mux function select. not used. can delete |

| 0x24    |    |      | func_io3        | register description            |
|---------|----|------|-----------------|---------------------------------|
| [31:30] | RW | 2'h0 | reserved        |                                 |
| [29:24] | RW | 6'h0 | r_func_io15_sel | pad 15 full mux function select |
| [23:22] | RW | 2'h0 | reserved        |                                 |
| [21:16] | RW | 6'h0 | r_func_io14_sel | pad 14 full mux function select |



|         |    |      |                 |                                 |
|---------|----|------|-----------------|---------------------------------|
| [15:14] | RW | 2'h0 | reserved        |                                 |
| [13:8]  | RW | 6'h0 | r_func_io13_sel | pad 13 full mux function select |
| [7:6]   | RW | 2'h0 | reserved        |                                 |
| [5:0]   | RW | 6'h0 | r_func_io12_sel | pad 12 full mux function select |

| 0x28    |    |      | func_io4        | register description            |
|---------|----|------|-----------------|---------------------------------|
| [31:30] | RW | 2'h0 | reserved        |                                 |
| [29:24] | RW | 6'h0 | r_func_io19_sel | pad 19 full mux function select |
| [23:22] | RW | 2'h0 | reserved        |                                 |
| [21:16] | RW | 6'h0 | r_func_io18_sel | pad 18 full mux function select |
| [15:14] | RW | 2'h0 | reserved        |                                 |
| [13:8]  | RW | 6'h0 | r_func_io17_sel | pad 17 full mux function select |
| [7:6]   | RW | 2'h0 | reserved        |                                 |
| [5:0]   | RW | 6'h0 | r_func_io16_sel | pad 16 full mux function select |

| 0x2c    |    |      | func_io5        | register description            |
|---------|----|------|-----------------|---------------------------------|
| [31:30] | RW | 2'h0 | reserved        |                                 |
| [29:24] | RW | 6'h0 | r_func_io23_sel | pad 23 full mux function select |
| [23:22] | RW | 2'h0 | reserved        |                                 |
| [21:16] | RW | 6'h0 | r_func_io22_sel | pad 22 full mux function select |
| [15:14] | RW | 2'h0 | reserved        |                                 |
| [13:8]  | RW | 6'h0 | r_func_io21_sel | pad 21 full mux function select |
| [7:6]   | RW | 2'h0 | reserved        |                                 |
| [5:0]   | RW | 6'h0 | r_func_io20_sel | pad 20 full mux function select |

| 0x30    |    |      | func_io6        | register description            |
|---------|----|------|-----------------|---------------------------------|
| [31:30] | RW | 2'h0 | reserved        |                                 |
| [29:24] | RW | 6'h0 | r_func_io27sel  | pad 27 full mux function select |
| [23:22] | RW | 2'h0 | reserved        |                                 |
| [21:16] | RW | 6'h0 | r_func_io26_sel | pad 26 full mux function select |
| [15:14] | RW | 2'h0 | reserved        |                                 |
| [13:8]  | RW | 6'h0 | r_func_io25_sel | pad 25 full mux function select |
| [7:6]   | RW | 2'h0 | reserved        |                                 |
| [5:0]   | RW | 6'h0 | r_func_io24_sel | pad 24 full mux function select |

| 0x34    |    |      | func_io7        | register description            |
|---------|----|------|-----------------|---------------------------------|
| [31:30] | RW | 2'h0 | reserved        |                                 |
| [29:24] | RW | 6'h0 | r_func_io31sel  | pad 31 full mux function select |
| [23:22] | RW | 2'h0 | reserved        |                                 |
| [21:16] | RW | 6'h0 | r_func_io30_sel | pad 30 full mux function select |
| [15:14] | RW | 2'h0 | reserved        |                                 |
| [13:8]  | RW | 6'h0 | r_func_io29_sel | pad 29 full mux function select |

|       |    |      |                 |                                 |
|-------|----|------|-----------------|---------------------------------|
| [7:6] | RW | 2'h0 | reserved        |                                 |
| [5:0] | RW | 6'h0 | r_func_io28_sel | pad 28 full mux function select |

| 0x38    |    |       | func_io8        | register description            |
|---------|----|-------|-----------------|---------------------------------|
| [31:22] | RW | 10'h0 | reserved        |                                 |
| [21:16] | RW | 6'h0  | r_func_io34_sel | pad 34 full mux function select |
| [15:14] | RW | 2'h0  | reserved        |                                 |
| [13:8]  | RW | 6'h0  | r_func_io33_sel | pad 33 full mux function select |
| [7:6]   | RW | 2'h0  | reserved        |                                 |
| [5:0]   | RW | 6'h0  | r_func_io32_sel | pad 32 full mux function select |

| 0x4C    |    |       | key_scan_in_en | register description |
|---------|----|-------|----------------|----------------------|
| [31:16] | RW | 16'h0 | reserved       |                      |
| [15:0]  | RW | 16'h0 | r_kscan_in_en  | key scan in enable   |

| 0x50    |    |       | key_scan_out_en | register description |
|---------|----|-------|-----------------|----------------------|
| [31:18] | RW | 14'h0 | reserved        |                      |
| [17:0]  | RW | 18'h0 | r_kscan_out_en  | key scan out enable  |

Physical IO PAD control registers:

Base address: 4000\_F000.

| 0xF008    |    |        | IOCTL0   |
|-----------|----|--------|--|
| [31 : 30] | RW | 2'd0   | pull up/down control of pin 09<br>00: floating, no pull up and pull down   |
| [29 : 28] | RW | 2'b0   | 01: weak pull up<br>10: strong pull up<br>11: pull down  |
| [27]      | RW | 1'b0   | wake up polarity select of pin 09<br>0: active POSEDGE<br>1: active NEGEDGE                                      |
| [26 : 24] | RW | 3'b110 | P08 is used for test mode config pin<br>pull up/down control of pin 07<br>00: floating, no pull up and pull down |
| [23 : 22] | RW | 2'b0   | 01: weak pull up<br>10: strong pull up<br>11: pull down  |
| [21]      | RW | 1'b0   | wake up polarity select of pin 07<br>0: active POSEDGE<br>1: active NEGEDGE                                      |

|           |    |       |   |
|-----------|----|-------|---|
| [20 : 19] | RW | 2'b0  | pull up/down control of pin 06<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [18]      | RW | 1'b0  | wake up polarity select of pin 06<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [17 : 16] | RW | 2'b0  | pull up/down control of pin 05<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [15]      | RW | 1'b0  | wake up polarity select of pin 05<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [14 : 13] | RW | 2'b0  | pull up/down control of pin 04<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [12]      | RW | 1'b0  | wake up polarity select of pin 04<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [11 : 10] | RW | 2'b11 | pull up/down control of pin 03<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [9]       | RW | 1'b0  | wake up polarity select of pin 03<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [ 8 : 7]  | RW | 2'b0  | pull up/down control of pin 02<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [6]       | RW | 1'b0  | wake up polarity select of pin 02<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [ 5 : 4]  | RW | 2'b0  | pull up/down control of pin 01<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [3]       | RW | 1'b0  | wake up polarity select of pin 01<br>0: active POSEDGE<br>1: active NEGEDGE   |

|          |    |      |   |
|----------|----|------|---|
| [ 2 : 1] | RW | 2'b0 | pull up/down control of pin 00<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [0]      | RW | 1'b0 | wake up polarity select of pin 00<br>0: active POSEDGE<br>1: active NEGEDGE   |

0xF00C

IOCTL1

|           |    |      |   |
|-----------|----|------|---|
| [31 : 30] | RW | 2'd0 |   |
| [29 : 28] | RW | 2'b0 | pull up/down control of pin 19<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [27]      | RW | 1'b0 | wake up polarity select of pin 19<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [26 : 25] | RW | 2'b0 | pull up/down control of pin 18<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [24]      | RW | 1'b0 | wake up polarity select of pin 18<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [23 : 22] | RW | 2'b0 | pull up/down control of pin 17<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [21]      | RW | 1'b0 | wake up polarity select of pin 17<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [20 : 19] | RW | 2'b0 | pull up/down control of pin 16<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [18]      | RW | 1'b0 | wake up polarity select of pin 16<br>0: active POSEDGE<br>1: active NEGEDGE   |

|           |    |      |   |
|-----------|----|------|---|
| [17 : 16] | RW | 2'b0 | pull up/down control of pin 15<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [15]      | RW | 1'b0 | wake up polarity select of pin 15<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [14 : 13] | RW | 2'b0 | pull up/down control of pin 14<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [12]      | RW | 1'b0 | wake up polarity select of pin 14<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [11 : 10] | RW | 2'b0 | pull up/down control of pin 13<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [9]       | RW | 1'b0 | wake up polarity select of pin 13<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [ 8 : 7]  | RW | 2'b0 | pull up/down control of pin 12<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [6]       | RW | 1'b0 | wake up polarity select of pin 12<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [ 5 : 4]  | RW | 2'b0 | pull up/down control of pin 11<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [3]       | RW | 1'b0 | wake up polarity select of pin 11<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [ 2 : 1]  | RW | 2'b0 | pull up/down control of pin 10<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [0]       | RW | 1'b0 | wake up polarity select of pin 10<br>0: active POSEDGE<br>1: active NEGEDGE   |

| 0xF010    |    | IOCTL2 |   |
|-----------|----|--------|---|
| [31 : 30] | RW | 2'd0   | pull up/down control of pin 29<br>00: floating, no pull up and pull down  |
| [29 : 28] | RW | 2'b0   | 01: weak pull up<br>10: strong pull up<br>11: pull down   |
| [27]      | RW | 1'b0   | wake up polarity select of pin 29<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [26 : 25] | RW | 2'b0   | pull up/down control of pin 28<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [24]      | RW | 1'b0   | wake up polarity select of pin 28<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [23 : 22] | RW | 2'b0   | pull up/down control of pin 27<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [21]      | RW | 1'b0   | wake up polarity select of pin 27<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [20 : 19] | RW | 2'b0   | pull up/down control of pin 26<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [18]      | RW | 1'b0   | wake up polarity select of pin 26<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [17 : 16] | RW | 2'b11  | pull up/down control of pin 25<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [15]      | RW | 1'b0   | wake up polarity select of pin 25<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [14 : 13] | RW | 2'b11  | pull up/down control of pin 24<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |

|           |    |      |   |
|-----------|----|------|---|
| [12]      | RW | 1'b0 | wake up polarity select of pin 24<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [11 : 10] | RW | 2'b0 | pull up/down control of pin 23<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [9]       | RW | 1'b0 | wake up polarity select of pin 23<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [ 8 : 7]  | RW | 2'b0 | pull up/down control of pin 22<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [6]       | RW | 1'b0 | wake up polarity select of pin 22<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [ 5 : 4]  | RW | 2'b0 | pull up/down control of pin 21<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [3]       | RW | 1'b0 | wake up polarity select of pin 21<br>0: active POSEDGE<br>1: active NEGEDGE   |
| [ 2 : 1]  | RW | 2'b0 | pull up/down control of pin 20<br>00: floating, no pull up and pull down<br>01: weak pull up<br>10: strong pull up<br>11: pull down |
| [0]       | RW | 1'b0 | wake up polarity select of pin 20<br>0: active POSEDGE<br>1: active NEGEDGE   |

### 3.10 GPIO

The General Purpose I/Os are a type of peripheral that can be mapped to physical I/O pads and programmed by software. The flexible GPIO are organized as two PORTs. Among them, PortA has bi-direction 18 bit lines, e.g., GPIO\_PORTA[17:0], while PortB has 17 bi-directional bit lines, e.g., PIO\_PORTB[16:0]. With default setting, physical pads: P00-P17 are connected to to PortA; Pads P18-34 are connected to PortB, when all GPIOs are enabled, as described in the IOMUX table in IOMUX section.

All PortA and PortB pins can be configured as bi-directional serial interface, by selecting as input or output direction, and their corresponding data can be either read from or written to registers. All PortA and PortB pins support wake-up, but only 18 PortA pins support interrupt. Also only PortA pins support debounce function.

Each GPIO pins can be pulled up to AVDD33 or pulled down to ground by adding pull up or pull down resistors to have default functions/states.

For more detailed info, please refer to “PHY62xx GPIO Application Notes”, in software SDK document folder. Below table are the Registers related to GPIOs.

Base address: 0x4000\_8000

| OFFSET  | TYPE | RESET | NAME                           | DESCRIPTION  |
|---------|------|-------|--------------------------------|--|
| 0x00    |      |       | gpio_swporta_dr                |  |
| [31:18] | RO   | 14'b0 | Reserved                       | Reserved   |
| [17:0]  | RW   | 18'b0 | Port A Data Register           | Values written to this register are output on the I/O signals for Port A   |
| 0x04    |      |       | gpio_swporta_ddr               |  |
| [31:18] | RO   | 14'b0 | Reserved                       | Reserved   |
| [17:0]  | RW   | 18'b0 | Port A Data Direction Register | Values written to this register independently control the direction of the corresponding data bit in Port A<br>1'b0: Input<br>1'b1: Output               |
| 0x08    |      |       | gpio_swporta_ctl               |  |
| [31:1]  | RO   | 31'b0 | Reserved                       | Reserved   |
| [0]     | RW   | 1'b0  | Port A Data Source             | The data and control source for a signal can come from either software or hardware<br>1'b0: Software mode<br>1'b1: Hardware mode                         |
| 0x0c    |      |       | gpio_swportb_dr                |  |
| [31:15] | RO   | 15'b0 | Reserved                       | Reserved   |
| [16:0]  | RW   | 17'b0 | Port B Data Register           | Values written to this register are output on the I/O signals for Port B   |
| 0x10    |      |       | gpio_swportb_ddr               |  |
| [31:15] | RO   | 15'b0 | Reserved                       | Reserved   |
| [16:0]  | RW   | 17'b0 | Port B Data Direction Register | Values written to this register independently control the direction of the corresponding data bit in Port B<br>1'b0: Input<br>1'b1: Output               |
| 0x14    |      |       | gpio_swportb_ctl               |  |
| [31:1]  | RO   | 31'b0 | Reserved                       | Reserved   |
| [0]     | RW   | 1'b0  | Port B Data Source             | The data and control source for a signal can come from either software or hardware<br>1'b0: Software mode<br>1'b1: Hardware mode                         |
| 0x30    |      |       | gpio_inten                     |  |
| [31:18] | RO   | 14'b0 | Reserved                       | Reserved   |
| [17:0]  | RW   | 18'b0 | Interrupt enable               | Allows each bit of Port A to be configured for interrupts<br>1'b0: Configure Port A bit as normal GPIO signal<br>1'b1: Configure Port A bit as interrupt |
| 0x34    |      |       | gpio_intmask                   |  |



|         |    |       |                      |   |
|---------|----|-------|----------------------|---|
| [31:18] | RO | 14'b0 | Reserved             | Reserved  |
| [17:0]  | RW | 18'b0 | Interrupt mask       | Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it<br>1'b0: Interrupt bits are unmasked<br>1'b1: Mask interrupt                                 |
| 0x38    |    |       | gpio_inttype_level   |   |
| [31:18] | RO | 14'b0 | Reserved             | Reserved  |
| [17:0]  | RW | 18'b0 | Interrupt level      | Controls the type of interrupt that can occur on Port A<br>1'b0: Level-sensitive<br>1'b1: Edge-sensitive  |
| 0x3c    |    |       | gpio_int_polarity    |   |
| [31:18] | RO | 14'b0 | Reserved             | Reserved  |
| [17:0]  | RW | 18'b0 | Interrupt polarity   | Controls the polarity of edge or level sensitivity that can occur on input of Port A<br>1'b0: Active-low or falling-edge<br>1'b1: Active-high or rising-edge  |
| 0x40    |    |       | gpio_intstatus       |   |
| [31:18] | RO | 14'b0 | Reserved             | Reserved  |
| [17:0]  | RO | 18'b0 | Interrupt status     | Interrupt status of Port A  |
| 0x44    |    |       | gpio_raw_intstatus   |   |
| [31:18] | RO | 14'b0 | Reserved             | Reserved  |
| [17:0]  | RO | 18'b0 | Raw interrupt status | Raw interrupt of status of Port A   |
| 0x48    |    |       | gpio_debounce        |   |
| [31:18] | RO | 14'b0 | Reserved             | Reserved  |
| [17:0]  | RW | 18'b0 | Debounce enable      | Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches<br>1'b0: No debounce<br>1'b1: Enable debounce                                  |
| 0x4c    |    |       | gpio_porta_eoi       |   |
| [31:18] | RO | 14'b0 | Reserved             | Reserved  |
| [17:0]  | WO | 18'b0 | Clear interrupt      | Controls the clearing of edge type interrupts from Port A<br>1'b0: No interrupt clear<br>1'b1: Clear interrupt  |
| 0x50    |    |       | gpio_ext_porta       |   |
| [31:18] | RO | 14'b0 | Reserved             | Reserved  |
| [17:0]  | RO | 18'b0 | External Port A      | When Port A is configured as Input, then reading this location reads the values on the signal. When the data direction of Port A is set as Output, reading this location reads the data register for Port A |
| 0x54    |    |       | gpio_ext_portb       |   |
| [31:17] | RO | 15'b0 | Reserved             | Reserved  |
| [16:0]  | RO | 17'b0 | External Port B      | When Port B is configured as Input, then reading this location reads the values on the signal. When the data direction of Port B is set as Output, reading this location reads the data register for Port B |
| 0x60    |    |       | gpio_ls_sync         |   |
| [31:1]  | RO | 31'b0 | Reserved             | Reserved  |

|         |    |         |                        |   |
|---------|----|---------|------------------------|---|
| [0]     | RW | 1'b0    | Synchronization level  | Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr<br>1'b0: No synchronization to pclk_intr<br>1'b1: Synchronize to pclk_intr |
| 0x64    |    |         | gpio_id_code           |   |
| [31:16] | RO | 16'b0   | Reserved               | Reserved  |
| [15:0]  | RO | 16'b0   | GPIO ID code           | This is a user-specified code that a system can read. It can be used for chip identification, and so on   |
| 0x6c    |    |         | gpio_ver_id_code       |   |
| [31:0]  | RO | 32'b0   | GPIO Component Version | ASCII value for each number in the version  |
| 0x74    |    |         | gpio_config_reg1       |   |
| [31:21] | RO | 11'b0   | Reserved               | Reserved  |
| [20:16] | RO | 5'b0x0f | ENCODED_ID_WIDTH       | The value of this register is equal to GPIO_ID_WIDTH-1  |
| [15]    | RO | 1'b0    | GPIO_ID                | The value of this register is derived from the GPIO_ID configuration parameter<br>1'b0: Exclude<br>1'b1: Include  |
| [14]    | RO | 1'b0    | ADD_ENCODED_PARAMS     | The value of this register is derived from the GPIO_ADD_ENCODED_PARAMS configuration parameter<br>1'b0: False<br>1'b1: True   |
| [13]    | RO | 1'b0    | DEBOUNCE               | The value of this register is derived from the GPIO_DEBOUNCE configuration parameter<br>1'b0: Exclude<br>1'b1: Include  |
| [12]    | RO | 1'b0    | PORTA_INTR             | The value of this register is derived from the GPIO_PORTA_INTR configuration parameter<br>1'b0: Exclude<br>1'b1: Include  |
| [11]    | RO | 1'b0    | Reserved               | Reserved  |
| [10]    | RO | 1'b0    | Reserved               | Reserved  |
| [9]     | RO | 1'b0    | HW_PORTB               | The value of this register is derived from the GPIO_HW_PORTB configuration parameter<br>1'b0: Exclude<br>1'b1: Include  |
| [8]     | RO | 1'b0    | HW_PORTA               | The value of this register is derived from the GPIO_HW_PORTA configuration parameter<br>1'b0: Exclude<br>1'b1: Include  |
| [7]     | RO | 1'b0    | Reserved               | Reserved  |
| [6]     | RO | 1'b0    | Reserved               | Reserved  |
| [5]     | RO | 1'b0    | PORTB_SINGLE_CTL       | The value of this register is derived from the GPIO_PORTB_SINGLE_CTL configuration parameter<br>1'b0: False<br>1'b1: True   |
| [4]     | RO | 1'b0    | PORTA_SINGLE_CTL       | The value of this register is derived from the GPIO_PORTA_SINGLE_CTL configuration parameter<br>1'b0: False<br>1'b1: True   |

|         |    |         |                     |  |
|---------|----|---------|---------------------|--|
| [3:2]   | RO | 2'b0x2  | NUM_PORTS           | The value of this register is derived from the GPIO_NUM_PORT configuration parameter<br>2'b00 1<br>2'b01 2<br>2'b10 3<br>2'b11 4                               |
| [1:0]   | RO | 2'b0x2  | APB_DATA_WIDTH      | The value of this register is derived from the GPIO_APB_DATA_WIDTH configuration parameter<br>2'b00 8 bits<br>2'b01 16 bits<br>2'b10 32 bits<br>2'b11 Reserved |
| 0x70    |    |         | gpio_config_reg2    |  |
| [31:10] | RO | 22'b0   | Reserved            | Reserved   |
| [9:5]   | RO | 5'b0x0f | ENCODED_ID_PWIDTH_B | The value of this register is equal to GPIO_PWIDTH_B-1   |
| [4:0]   | RO | 5'b0x11 | ENCODED_ID_PWIDTH_A | The value of this register is equal to GPIO_PWIDTH_A-1   |

## 4 Peripheral Blocks

### 4.1 2.4GHz Radio

The 2.4 GHz RF transceiver is designed to operate in the worldwide ISM frequency band at 2.4 to 2.4835 GHz. Radio modulation modes and configurable packet structure make the transceiver interoperable with *Bluetooth*® low energy (BLE) protocol implementations.

- General modulation format
  - FSK (configurable modulation index) with configurable Gaussian Filter Shaping
  - OQPSK with half-sine shaping
  - On-air data rates
    - 125kbps/250kbps/500kbps/1Mbps/2Mbps
- Transmitter with programmable output power of -20dBm to +10dBm, in 3dB steps
- RSSI function (1 dB resolution, ± 2 dB accuracy)
- Receiver sensitivity
  - -103dBm@125Kbps GFSK
  - -98dBm@500Kbps GFSK
  - -97dBm@1Mbps BLE
  - -94dBm@2Mbps BLE
- Embedded RF balun
- Integrated frac-N synthesizer with phase modulation

### 4.2 Timer/Counters (TIMER)

The implementation can include a 24-bit SysTick system timer, that extends the functionality of both the processor and the NVIC. When present, the NVIC part of the extension provides:

- A 24-bit system timer (SysTick)
- Additional configurable priority SysTick interrupt.

- See the ARMv7-M ARM for more information.

General purpose timers are included in the design. This timer is Synopsys DW\_apb\_timer. With the input clock running at 4Mhz.

The timer related registers are listed below, and there are two sets of identical timers.

Base address: Timer setA: 4000\_1000, timer\_setB: 4002\_1000

| OFFSET  | TYPE | RESET | NAME                             | DESCRIPTION  |
|---------|------|-------|----------------------------------|--|
| 0x00    |      |       | Timer1LoadCount                  |  |
| [31:24] | RO   | 8'b0  | Reserved                         | Reserved   |
| [23:0]  | RW   | 24'b0 | Timer1 Load Count Register       | Value to be loaded into Timer1   |
| 0x04    |      |       | Timer1CurrentValue               |  |
| [31:24] | RO   | 8'b0  | Reserved                         | Reserved   |
| [23:0]  | RO   | 24'b0 | Timer1 Current Value Register    | Current Value of Timer1  |
| 0x08    |      |       | Timer1ControlReg                 |  |
| [31:3]  | RO   | 29'b0 | Reserved                         | Reserved   |
| [2]     | RW   | 1'b0  | Timer Interrupt Mask             | Timer interrupt mask for Timer1<br>1'b0: not masked<br>1'b1: masked                    |
| [1]     | RW   | 1'b0  | Timer Mode                       | Timer mode for Timer1<br>1'b0: free-running mode<br>1'b1: user-defined count mode      |
| [0]     | RW   | 1'b0  | Timer Enable                     | Timer enable bit for Timer1<br>1'b0: disable<br>1'b1: enable                           |
| 0x0c    |      |       | Timer1EOI                        |  |
| [31:1]  | RO   | 31'b0 | Reserved                         | Reserved   |
| [0]     | RO   | 1'b0  | Timer1 End of-Interrupt Register | Reading from this register returns all zeroes (0) and clears the interrupt from Timer1 |
| 0x10    |      |       | Timer1IntStatus                  |  |
| [31:1]  | RO   | 31'b0 | Reserved                         | Reserved   |
| [0]     | RO   | 1'b0  | Timer1 Interrupt Status Register | Contains the interrupt status for Timer1   |
| 0x14    |      |       | Timer2LoadCount                  |  |
| [31:24] | RO   | 8'b0  | Reserved                         | Reserved   |
| [23:0]  | RW   | 24'b0 | Timer2 Load Count Register       | Value to be loaded into Timer2   |
| 0x18    |      |       | Timer2CurrentValue               |  |
| [31:24] | RO   | 8'b0  | Reserved                         | Reserved   |
| [23:0]  | RO   | 24'b0 | Timer2 Current Value Register    | Current Value of TimerN  |
| 0x1c    |      |       | Timer2ControlReg                 |  |
| [31:3]  | RO   | 29'b0 | Reserved                         | Reserved   |
| [2]     | RW   | 1'b0  | Timer Interrupt Mask             | Timer interrupt mask for Timer2<br>1'b0: not masked<br>1'b1: masked                    |

|         |    |                    |                                  |  |
|---------|----|--------------------|----------------------------------|--|
| [1]     | RW | 1'b0               | Timer Mode                       | Timer mode for Timer2<br>1'b0: free-running mode<br>1'b1: user-defined count mode      |
| [0]     | RW | 1'b0               | Timer Enable                     | Timer enable bit for Timer2<br>1'b0: disable<br>1'b1: enable                           |
| 0x20    |    | Timer2EOI          |                                  |  |
| [31:1]  | RO | 31'b0              | Reserved                         | Reserved   |
| [0]     | RO | 1'b0               | Timer2 End of-Interrupt Register | Reading from this register returns all zeroes (0) and clears the interrupt from Timer2 |
| 0x24    |    | Timer2IntStatus    |                                  |  |
| [31:1]  | RO | 31'b0              | Reserved                         | Reserved   |
| [0]     | RO | 1'b0               | Timer2 Interrupt Status Register | Contains the interrupt status for Timer2   |
| 0x28    |    | Timer3LoadCount    |                                  |  |
| [31:24] | RO | 8'b0               | Reserved                         | Reserved   |
| [23:0]  | RW | 24'b0              | Timer3 Load Count Register       | Value to be loaded into Timer3   |
| 0x2c    |    | Timer3CurrentValue |                                  |  |
| [31:24] | RO | 8'b0               | Reserved                         | Reserved   |
| [23:0]  | RO | 24'b0              | Timer3 Current Value Register    | Current Value of TimerN  |
| 0x30    |    | Timer3ControlReg   |                                  |  |
| [31:3]  | RO | 29'b0              | Reserved                         | Reserved   |
| [2]     | RW | 1'b0               | Timer Interrupt Mask             | Timer interrupt mask for Timer3<br>1'b0: not masked<br>1'b1: masked                    |
| [1]     | RW | 1'b0               | Timer Mode                       | Timer mode for Timer3<br>1'b0: free-running mode<br>1'b1: user-defined count mode      |
| [0]     | RW | 1'b0               | Timer Enable                     | Timer enable bit for Timer3<br>1'b0: disable<br>1'b1: enable                           |
| 0x34    |    | Timer3EOI          |                                  |  |
| [31:1]  | RO | 31'b0              | Reserved                         | Reserved   |
| [0]     | RO | 1'b0               | Timer3 End of-Interrupt Register | Reading from this register returns all zeroes (0) and clears the interrupt from Timer3 |
| 0x38    |    | Timer3IntStatus    |                                  |  |
| [31:1]  | RO | 31'b0              | Reserved                         | Reserved   |
| [0]     | RO | 1'b0               | Timer3 Interrupt Status Register | Contains the interrupt status for Timer3   |
| 0x3c    |    | Timer4LoadCount    |                                  |  |
| [31:24] | RO | 8'b0               | Reserved                         | Reserved   |
| [23:0]  | RW | 24'b0              | Timer4 Load Count Register       | Value to be loaded into Timer4   |
| 0x40    |    | Timer4CurrentValue |                                  |  |
| [31:24] | RO | 8'b0               | Reserved                         | Reserved   |
| [23:0]  | RO | 24'b0              | Timer4 Current Value Register    | Current Value of Timer4  |

|        |    |       |                                      |  |
|--------|----|-------|--------------------------------------|--|
| 0x44   |    |       | Timer4ControlReg                     |  |
| [31:3] | RO | 29'b0 | Reserved                             | Reserved   |
| [2]    | RW | 1'b0  | Timer Interrupt Mask                 | Timer interrupt mask for Timer4<br>1'b0: not masked<br>1'b1: masked  |
| [1]    | RW | 1'b0  | Timer Mode                           | Timer mode for Timer4<br>1'b0: free-running mode<br>1'b1: user-defined count mode  |
| [0]    | RW | 1'b0  | Timer Enable                         | Timer enable bit for Timer4<br>1'b0: disable<br>1'b1: enable   |
| 0x48   |    |       | Timer4EOI                            |  |
| [31:1] | RO | 31'b0 | Reserved                             | Reserved   |
| [0]    | RO | 1'b0  | Timer4 End of-Interrupt Register     | Reading from this register returns all zeroes (0) and clears the interrupt from Timer4   |
| 0x4c   |    |       | Timer4IntStatus                      |  |
| [31:1] | RO | 31'b0 | Reserved                             | Reserved   |
| [0]    | RO | 1'b0  | Timer4 Interrupt Status Register     | Contains the interrupt status for Timer4   |
| 0xa0   |    |       | TimersIntStatus                      |  |
| [31:4] | RO | 28'b0 | Reserved                             | Reserved   |
| [3:0]  | RO | 4'b0  | Timers Interrupt Status Register     | Contains the interrupt status of all timers in the component<br>0: either timer_intr or timer_intr_n is not active after masking<br>1: either timer_intr or timer_intr_n is active after masking                             |
| 0xa4   |    |       | TimersEOI                            |  |
| [31:4] | RO | 28'b0 | Reserved                             | Reserved   |
| [3:0]  | RO | 4'b0  | Timers End of-Interrupt Register     | Reading this register returns all zeroes (0) and clears all active interrupts  |
| 0xa8   |    |       | TimersRawIntStatus                   |  |
| [31:4] | RO | 28'b0 | Reserved                             | Reserved   |
| [3:0]  | RO | 4'b0  | Timers Raw Interrupt Status Register | The register contains the unmasked interrupt status of all timers in the component<br>0: either timer_intr or timer_intr_n is not active prior to masking<br>1: either timer_intr or timer_intr_n is active prior to masking |
| 0xac   |    |       | TimersRawIntStatus                   |  |
| [31:0] | RO | 32'b0 | Timers Component Version             | Current revision number of the DW_apb_timers component   |

### 4.3 Real Time Counter (RTC)

The Real Time Counter (RTC) module provides a generic, low power timer on the low-frequency clock source (LFCLK). The RTC features a 24 bit COUNTER, 12 bit (1/X) prescaler, capture/compare registers, and a tick event generator for low power, tickless RTOS implementation.

RTC related registers are listed below:

Base address: 4000\_F000

| 0xF024    |    | RTCCTL |  |
|-----------|----|--------|--|
| [31 : 24] | RW | 8'h0   |  |
|           | RW | 1'b0   | Counter overflow event enable.   |
| [23]      |    |        | 1'b0: disable<br>1'b1: enable  |
|           | RW | 1'b0   | Comparator 2 event enable.   |
| [22]      |    |        | 1'b0: disable<br>1'b1: enable  |
|           | RW | 1'b0   | Comparator 1 event enable.   |
| [21]      |    |        | 1'b0: disable<br>1'b1: enable  |
|           | RW | 1'b0   | Comparator 0 event enable.   |
| [20]      |    |        | 1'b0: disable<br>1'b1: enable  |
|           | RW | 1'b0   | RTC tick event enable.   |
| [19]      |    |        | 1'b0: disable<br>1'b1: enable  |
|           | RW | 1'b0   | Counter overflow interrupt enable.   |
| [18]      |    |        | 1'b0: disable<br>1'b1: enable  |
|           | RW | 1'b0   | Comparator 2 interrupt enable.   |
| [17]      |    |        | 1'b0: disable<br>1'b1: enable  |
|           | RW | 1'b0   | Comparator 1 interrupt enable.   |
| [16]      |    |        | 1'b0: disable<br>1'b1: enable  |
|           | RW | 1'b0   | Comparator 0 interrupt enable.   |
| [15]      |    |        | 1'b0: disable<br>1'b1: enable  |
|           | RW | 1'b0   | RTC tick interrupt enable.   |
| [14]      |    |        | 1'b0: disable<br>1'b1: enable  |
| [13 : 2]  | RW | 12'h0  | 12bit prescaler for RTC counter frequency<br>(32768/(PRESCALER+1)).Can be written only<br>when RTC is stopped.   |
| [1]       | RW | 1'b0   | RTC counter clear bit. Write 1'b1 will clear RTC<br>counter and after one clock this bit will return to<br>1'b0. |
| [0]       | RW | 1'b0   | RTC run/stop control.<br>1'b0: stop<br>1'b1: run   |

| 0xF028    |    | RTCCNT |  |
|-----------|----|--------|--|
| [31 : 24] | RO | 8'h0   |  |
|           | RO | 24'h0  | Writing 32'h5A5AA5A5 can trigger the overflow<br>task that sets the RTC counter value to<br>24'hFFFFFF0 to allow SW test of the overflow<br>condition. |
| [23 : 0]  |    |        |  |

Reading can read the value of RTC counter (low 24 bits).

|           |    |       |                                      |
|-----------|----|-------|--------------------------------------|
| 0xF02C    |    |       | RTCCC0                               |
| [31 : 24] | RW | 8'h0  |                                      |
| [23 : 0]  | RW | 24'h0 | Compare value of comparator 0        |
| 0xF030    |    |       | RTCCC1                               |
| [31 : 24] | RW | 8'h0  |                                      |
| [23 : 0]  | RW | 24'h0 | Compare value of comparator 1        |
| 0xF034    |    |       | RTCCC2                               |
| [31 : 24] | RW | 8'h0  |                                      |
| [23 : 0]  | RW | 24'h0 | Compare value of comparator 2        |
| 0xF038    |    |       | RTCFLAG                              |
| [31 : 4]  | R  | 28'h0 |                                      |
| [3]       | RO | 1'b0  | Overflow result flag.                |
| [2]       | RO | 1'b0  | Compare result flag of comparator 2. |
| [1]       | RO | 1'b0  | Compare result flag of comparator 1. |
| [0]       | RO | 1'b0  | Compare result flag of comparator 0. |

## 4.4 AES-ECB Encryption (ECB)

The ECB encryption block supports 128 bit AES encryption. It can be used for a range of cryptographic functions like hash generation, digital signatures, and keystream generation for data encryption/decryption.

AES-ECB related registers are listed below:

Base address:4004\_0000

| OFFSET  | TYPE | RESET                      | NAME              | DESCRIPTION  |
|---------|------|----------------------------|-------------------|--|
| 0x00    |      | AES layer enable register  |                   |  |
| [31:1]  | —    | 31'b0                      | reserved          |  |
| [0]     | RW   | 1'b0                       | Enable            | Setting this bit to “1” will enable AES to do TX/RX      |
| 0x04    |      | AES layer control register |                   |  |
| [31:17] | —    | 15'b0                      | reserved          |  |
| [16]    | RW   | 1'b0                       | Fifo out/in (PDU) | if pdu is little-endian set 0;if pdu is big-endian set 1 |
| [15:12] | —    | 4'b0                       | reserved          |  |



|   |    |       |                        |   |
|---|----|-------|------------------------|---|
|   |    |       |                        | [11]:data out:<br>if it is little-endian set 0<br>if it is big-endian set 1                 |
| [11:8]                                    | RW | 4'b0  | Enginne revert         | [10]:xor data :1<br>[9]: key :<br>if it is little-endian set 0<br>if it is big-endian set 1 |
|   |    |       |                        | [8]:data<br>if it is little-endian set 0<br>if it is big-endian set 1                       |
| [7:5]                                     | —  | 3'b0  | reserved               |   |
| [4]                                       | RW | 1'b0  | Aes_single mode        | Aes single mode   |
| [3]                                       | RW | 1'b0  | Code_mode              | Encrypt /decrypt  |
| [2:0]                                     | —  | 3'b0  | reserved               |   |
| <b>0x08 AES reserved register</b>         |    |       |                        |   |
| [31:0]                                    | —  | 32'b0 | reserved               |   |
| <b>0x0c AES plen &amp; aad register</b>   |    |       |                        |   |
| [31:16]                                   | —  | 16'b0 | reserved               |   |
| [15:8]                                    | RW | 8'b0  | plen                   | Packet length   |
| [7:0]                                     | RW | 8'b0  | aad                    | aad   |
| <b>0x10 AES interrupt mask register</b>   |    |       |                        |   |
| [31:4]                                    | —  | 28'b0 | reserved               |   |
| [3:0]                                     | RW | 4'b0  | Aes interupt<br>enable | [0]: encrypt done;[1]: decrypt failed;[2]:<br>decrypt ok;[3] single mode done               |
| <b>0x14 AES interrupt status register</b> |    |       |                        |   |
| [31:4]                                    | —  | 28'b0 | reserved               |   |
| [3:0]                                     | RO | 4'b0  | Aes interupt<br>status | [0]: encrypt done;[1]: decrypt failed;[2]:<br>decrypt ok;[3] single mode done               |
| <b>0x18 AES reserved register</b>         |    |       |                        |   |
| [31:0]                                    | —  | 32'b0 | reserved               |   |
| <b>0x1C AES reserved register</b>         |    |       |                        |   |
| [31:0]                                    | —  | 32'b0 | reserved               |   |
| <b>0x20 AES key0 register</b>             |    |       |                        |   |
| [31:0]                                    | RW | 32'b0 | Key0[31:0]             | Key[31:0]   |
| <b>0x24 AES key1 register</b>             |    |       |                        |   |
| [31:0]                                    | RW | 32'b0 | Key1[31:0]             | Key[63:32]  |
| <b>0x28 AES key2 register</b>             |    |       |                        |   |
| [31:0]                                    | RW | 32'b0 | Key2[31:0]             | Key[95:64]  |
| <b>0x2C AES key3 register</b>             |    |       |                        |   |
| [31:0]                                    | RW | 32'b0 | Key3[31:0]             | Key[127:96]   |

|        |    |       |               |  |
|--------|----|-------|---------------|--|
| 0x30   |    |       |               | AES nonce0 register  |
| [31:0] | RW | 32'b0 | Nonce0[31:0]  | Single mode:data_in[31:0];ECB-CCM:Nonce[31:0]                      |
| 0x34   |    |       |               | AES nonce1 register  |
| [31:0] | RW | 32'b0 | Nonce1[31:0]  | Single mode:data_in[31:0];ECB-CCM:Nonce[63:32]                     |
| 0x38   |    |       |               | AES nonce2 register  |
| [31:0] | RW | 32'b0 | Nonce2[31:0]  | Single mode:data_in[31:0];ECB-CCM:Nonce[95:64]                     |
| 0x3C   |    |       |               | AES nonce3 register  |
| [31:0] | RW | 32'b0 | Nonce3[31:0]  | Single mode:data_in[31:0];ECB-CCM:Nonce[127:96]                    |
| 0x50   |    |       |               | AES data out 0(single mode) register                               |
| [31:0] | RO | 32'b0 | Data_o0[31:0] | Data_out[31:0]   |
| 0x54   |    |       |               | AES data out 1(single mode) register                               |
| [31:0] | RO | 32'b0 | Data_o1[31:0] | Data_out[63:32]  |
| 0x58   |    |       |               | AES data out 2(single mode) register                               |
| [31:0] | RO | 32'b0 | Data_o2[31:0] | Data_out[95:64]  |
| 0x5C   |    |       |               | AES data out 3(single mode) register                               |
| [31:0] | RO | 32'b0 | Data_o3[31:0] | Data_out[127:96]   |
| 0x100  |    |       |               | AES memory (0x0100~0x01FC)   |
| [31:0] | RW | 32'b0 | memory write  | Writing offset address 0x100~0x1FC will write data into aes memory |

## 4.5 Random Number Generator (RNG)

The Random Number Generator (RNG) generates true non-deterministic random numbers based on internal thermal noise. These random numbers are suitable for cryptographic purposes. The RNG does not require a seed value.

## 4.6 Watchdog Timer (WDT)

A count down watchdog timer using the low-frequency clock source (LFCLK) offers configurable and robust protection against application lock-up. The watchdog can be paused during long CPU sleep periods for low power applications and when the debugger has halted the CPU.

## 4.7 SPI (SPI)

The SPI interface supports 3 serial synchronous protocols which are SPI, SSP and Microwire serial protocols. SPI wrapper contains one SPI master and one SPI slave. They are logically exclusive. Only one block is alive at a time. The operation mode for master mode and slave mode is controlled by

PERI\_MASTER\_SELECT Register in COM block.

| bit | Reset value | Definition                   |
|-----|-------------|------------------------------|
| 1   | 0           | SPI1 is master mode when set |
| 0   | 0           | SPI0 is master mode when set |

**Table 10: PERI\_MASTER\_SELECT Register bit definition  
(base address = 0x4000\_302C)**

SPI0 and SPI1 configuration registers are listed below:

Base address: SPI0: 4000\_6000; SPI1: 4000\_7000

|         |        |        |          |   |
|---------|--------|--------|----------|---|
| 0x00    | CTRLR0 |        |          |   |
| [31:16] | RO     | 16'b0  | Reserved | Reserved  |
| [15:12] | RW     | 4'b0   | CFS      | Control Frame Size. Selects the length of the control word for the Microwire frame format   |
| [11]    | RW     | 1'b0   | SRL      | Shift Register Loop. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input<br>1'b0: Normal Mode Operation<br>1'b1: Test Mode Operation  |
| [10]    | RW     | 1'b0   | SLV_OE   | Slave Output Enable<br>1'b0: Slave txd is enabled<br>1'b1: Slave txd is disabled  |
| [9:8]   | RW     | 2'b0   | TMOD     | Transfer Mode. Selects the mode of transfer for serial communication.<br>2'b00: Transmit & Receive<br>2'b01: Transmit Only<br>2'b10: Receive Only<br>2'b11: EEPROM Read   |
| [7]     | RW     | 1'b0   | SCPOL    | Serial Clock Polarity. Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock<br>1'b0: Inactive state of serial clock is low<br>1'b1: Inactive state of serial clock is high  |
| [6]     | RW     | 1'b0   | SCPH     | Serial Clock Phase. Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal<br>1'b0: Serial clock toggles in middle of first data bit<br>1'b1: Serial clock toggles at start of first data bit |
| [5:4]   | RW     | 2'b0   | FRF      | Frame Format. Selects which serial protocol transfers the data<br>2'b00: Motorola SPI<br>2'b01: Texas Instruments SSP<br>2'b10: National Semiconductors Microwire<br>2'b11: Reserved  |
| [3:0]   | RW     | 4'b0x7 | DFS      | Data Frame Size. Selects the data frame length  |
| 0x04    | CTRLR1 |        |          |   |
| [31:16] | RO     | 16'b0  | Reserved | Reserved  |
| [15:0]  | RW     | 16'b0  | NDF      | Number of Data Frames   |
| 0x08    | SSIENR |        |          |   |

|             |    |       |          |   |
|-------------|----|-------|----------|---|
| [31:1]      | RO | 31'b0 | Reserved | Reserved  |
| [0]         | RW | 1'b0  | SSI_EN   | This register enables and disables the DW_apb_ssi<br>1'b0: disable<br>1'b1: enable  |
| 0x0c MWCR   |    |       |          |   |
| [31:3]      | RO | 29'b0 | Reserved | Reserved  |
| [2]         | RW | 1'b0  | MHS      | Microwire Handshaking<br>1'b0: disabled<br>1'b1: enabled  |
| [1]         | RW | 1'b0  | MDD      | Microwire Control. Defines the direction of the data word when the Microwire serial protocol is used  |
| [0]         | RW | 1'b0  | MWMOD    | Microwire Transfer Mode. Defines whether the Microwire transfer is sequential or non-sequential<br>1'b0: non-sequential transfer<br>1'b1: sequential transfer |
| 0x10 SER    |    |       |          |   |
| [31:1]      | RO | 31'b0 | Reserved | Reserved  |
| [0]         | RW | 1'b0  | SER      | Slave Select Enable Flag<br>1'b0: non-sequential transfer<br>1'b1: sequential transfer  |
| 0x14 BAUDR  |    |       |          |   |
| [31:16]     | RO | 16'b0 | Reserved | Reserved  |
| [15:0]      | RW | 16'b0 | SCKDV    | SSI Clock Divider   |
| 0x18 TXFTLR |    |       |          |   |
| [31:3]      | RO | 29'b0 | Reserved | Reserved  |
| [2:0]       | RW | 3'b0  | TFT      | Transmit FIFO Threshold. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt                                 |
| 0x1c RXFTLR |    |       |          |   |
| [31:3]      | RO | 29'b0 | Reserved | Reserved  |
| [2:0]       | RW | 3'b0  | RFT      | Receive FIFO Threshold. Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt                                   |
| 0x20 TXFLR  |    |       |          |   |
| [31:4]      | RO | 28'b0 | Reserved | Reserved  |
| [3:0]       | RO | 4'b0  | TXTFL    | Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO   |
| 0x24 RXFLR  |    |       |          |   |
| [31:4]      | RO | 28'b0 | Reserved | Reserved  |
| [3:0]       | RO | 4'b0  | RXTFL    | Receive FIFO Level. Contains the number of valid data entries in the receive FIFO   |
| 0x28 SR     |    |       |          |   |
| [31:7]      | RO | 25'b0 | Reserved | Reserved  |
| [6]         | RO | 1'b0  | DCOL     | Data Collision Error<br>1'b0: No error<br>1'b1: Transmit data collision error   |
| [5]         | RO | 1'b0  | TXE      | Transmission Error. Set if the transmit FIFO is empty when a transfer is started<br>1'b0: No error<br>1'b1: Transmission error                                |

|        |    |       |          |   |
|--------|----|-------|----------|---|
| [4]    | RO | 1'b0  | RFF      | Receive FIFO Full<br>1'b0: not full<br>1'b1: full   |
| [3]    | RO | 1'b0  | RFNE     | Receive FIFO Not Empty<br>1'b0: empty<br>1'b1: not empty  |
| [2]    | RO | 1'b1  | TFE      | Transmit FIFO Empty<br>1'b0: not empty<br>1'b1: empty   |
| [1]    | RO | 1'b1  | TFNF     | Transmit FIFO Not Full<br>1'b0: full<br>1'b1: not full  |
| [0]    | RO | 1'b0  | BUSY     | SSI Busy Flag<br>1'b0: DW_apb_ssi is idle or disabled<br>1'b1: DW_apb_ssi is actively transferring data |
| 0x2c   |    | IMR   |          |   |
| [31:6] | RO | 26'b0 | Reserved | Reserved  |
| [5]    | RW | 1'b1  | MSTIM    | Multi-Master Contention Interrupt Mask<br>1'b0: masked<br>1'b1: not masked                              |
| [4]    | RW | 1'b1  | RXFIM    | Receive FIFO Full Interrupt Mask<br>1'b0: masked<br>1'b1: not masked                                    |
| [3]    | RW | 1'b1  | RXOIM    | Receive FIFO Overflow Interrupt Mask<br>1'b0: masked<br>1'b1: not masked                                |
| [2]    | RW | 1'b1  | RXUIM    | Receive FIFO Underflow Interrupt Mask<br>1'b0: masked<br>1'b1: not masked                               |
| [1]    | RW | 1'b1  | TXOIM    | Transmit FIFO Overflow Interrupt Mask<br>1'b0: masked<br>1'b1: not masked                               |
| [0]    | RW | 1'b1  | TXEIM    | Transmit FIFO Empty Interrupt Mask<br>1'b0: masked<br>1'b1: not masked                                  |
| 0x30   |    | ISR   |          |   |
| [31:6] | RO | 26'b0 | Reserved | Reserved  |
| [5]    | RO | 1'b0  | MSTIS    | Multi-Master Contention Interrupt Status<br>1'b0: not active<br>1'b1: active                            |
| [4]    | RO | 1'b0  | RXFIS    | Receive FIFO Full Interrupt Status<br>1'b0: not active<br>1'b1: active                                  |
| [3]    | RO | 1'b0  | RXOIS    | Receive FIFO Overflow Interrupt Status<br>1'b0: not active<br>1'b1: active                              |
| [2]    | RO | 1'b0  | RXUIS    | Receive FIFO Underflow Interrupt Status<br>1'b0: not active<br>1'b1: active                             |
| [1]    | RO | 1'b0  | TXOIS    | Transmit FIFO Overflow Interrupt Status<br>1'b0: not active<br>1'b1: active                             |

|              |    |       |          |   |
|--------------|----|-------|----------|---|
| [0]          | RO | 1'b0  | TXEIS    | Transmit FIFO Empty Interrupt Status<br>1'b0: not active<br>1'b1: active                                      |
| 0x34 RISR    |    |       |          |   |
| [31:6]       | RO | 26'b0 | Reserved | Reserved  |
| [5]          | RO | 1'b0  | MSTIR    | Multi-Master Contention Raw Interrupt Status<br>1'b0: not active<br>1'b1: active                              |
| [4]          | RO | 1'b0  | RXFIR    | Receive FIFO Full Raw Interrupt Status<br>1'b0: not active<br>1'b1: active                                    |
| [3]          | RO | 1'b0  | RXOIR    | Receive FIFO Overflow Raw Interrupt Status<br>1'b0: not active<br>1'b1: active                                |
| [2]          | RO | 1'b0  | RXUIR    | Receive FIFO Underflow Raw Interrupt Status<br>1'b0: not active<br>1'b1: active                               |
| [1]          | RO | 1'b0  | TXOIR    | Transmit FIFO Overflow Raw Interrupt Status<br>1'b0: not active<br>1'b1: active                               |
| [0]          | RO | 1'b0  | TXEIR    | Transmit FIFO Empty Raw Interrupt Status<br>1'b0: not active<br>1'b1: active                                  |
| 0x38 TXOICR  |    |       |          |   |
| [31:1]       | RO | 31'b0 | Reserved | Reserved  |
| [0]          | RO | 1'b0  | TXOICR   | Clear Transmit FIFO Overflow Interrupt  |
| 0x3c RXOICR  |    |       |          |   |
| [31:1]       | RO | 31'b0 | Reserved | Reserved  |
| [0]          | RO | 1'b0  | RXOICR   | Clear Receive FIFO Overflow Interrupt   |
| 0x40 RXUICR  |    |       |          |   |
| [31:1]       | RO | 31'b0 | Reserved | Reserved  |
| [0]          | RO | 1'b0  | RXUICR   | Clear Receive FIFO Underflow Interrupt  |
| 0x44 MSTICR  |    |       |          |   |
| [31:1]       | RO | 31'b0 | Reserved | Reserved  |
| [0]          | RO | 1'b0  | MSTICR   | Clear Multi-Master Contention Interrupt   |
| 0x48 ICR     |    |       |          |   |
| [31:1]       | RO | 31'b0 | Reserved | Reserved  |
| [0]          | RO | 1'b0  | ICR      | Clear Interrupts  |
| 0x4c DMACR   |    |       |          |   |
| [31:2]       | RO | 30'b0 | Reserved | Reserved  |
| [1]          | RW | 1'b0  | TDMAE    | Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel<br>1'b0: disable<br>1'b1: enable |
| [0]          | RW | 1'b0  | RDMAE    | Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel<br>1'b0: disable<br>1'b1: enable   |
| 0x50 DMATDLR |    |       |          |   |

|           |    |       |                  |  |
|-----------|----|-------|------------------|--|
| [31:3]    | RO | 29'b0 | Reserved         | Reserved   |
| [2:0]     | RW | 3'b0  | DMATDL           | Transmit Data Level  |
| 0x54      |    |       | DMARDLR          |  |
| [31:3]    | RO | 29'b0 | Reserved         | Reserved   |
| [2:0]     | RW | 3'b0  | DMARDL           | Receive Data Level   |
| 0x58      |    |       | IDR              |  |
| [31:0]    | RO | 32'b0 | IDCODE           | Identification Code  |
| 0x5c      |    |       | SSI_COMP_VERSION |  |
| [31:0]    | RO | 32'b0 | SSI_COMP_VERSION | Contains the hex representation of the Synopsys component version                                  |
| 0x60~0x9c |    |       | DR               |  |
| [31:16]   | RO | 16'b0 | Reserved         | Reserved   |
| [15:0]    | RW | 16'b0 | DR               | Data Register<br>Read: Receive FIFO buffer<br>Write: Transmit FIFO buffer                          |
| 0xf4      |    |       | RSVD_0           |  |
| [31:0]    | RW | 32'b0 | Reserved         | Reserved location for future use   |
| 0xf8      |    |       | RSVD_1           |  |
| [31:0]    | RW | 32'b0 | Reserved         | Reserved location for future use   |
| 0xfc      |    |       | RX_SAMPLE_DLY    |  |
| [31:8]    | RO | 24'b0 | Reserved         | Reserved   |
| [7:0]     | RW | 8'b0  | RSD              | Receive Data (rxd) Sample Delay. This register is used to delay the sample of the rxd input signal |

## 4.8 I2C (I2c0, I2c1 Two Independent Instances)

This I2C block support 100Khz, and 400Khz modes. It also supports 7-bit address and 10-bit address. It has built-in configurable spike suppression function for both lines.

I2C registers are listed below:

Base address: I2C0: 4000\_5000, I2C1: 4000\_5800

| OFFSET | TYPE | RESET | NAME                 | DESCRIPTION   |
|--------|------|-------|----------------------|---|
| 0x00   |      |       | I2C Control Register |   |
| [31:7] | RO   | 25'b0 | Reserved             | Reserved  |
| [6]    | RW   | 1'b0  | IC_SLAVE_DISABLE     | This bit controls whether I2C has its slave disabled<br>1'b0: slave is enabled<br>1'b1: slave is disabled   |
| [5]    | RW   | 1'b1  | IC_RESTART_EN        | Determines whether RESTART conditions may be sent when acting as a master<br>1'b0: disable<br>1'b1: enable  |
| [4]    | RW   | 1'b1  | IC_10BITADDR_MASTER  | Controls whether the DW_apb_i2c starts its transfers in 7- or 10-bit addressing mode when acting as a master<br>1'b0: 7-bit addressing<br>1'b1: 10-bit addressing |

|   |    |           |                     |   |
|---|----|-----------|---------------------|---|
| [3]                                     | RW | 1'b1      | IC_10BITADDR_SLAVE  | When acting as a slave, this bit controls whether the DW_apb_i2c responds to 7- or 10-bit addresses<br>1'b0: 7-bit addressing<br>1'b1: 10-bit addressing  |
| [2:1]                                   | RW | 2'b11     | SPEED               | These bits control at which speed the DW_apb_i2c operates<br>2'b01: standard mode<br>2'b10: fast mode<br>2'b11: high speed mode   |
| [0]                                     | RW | 1'b0      | MASTER_MODE         | This bit controls whether the DW_apb_i2c master is enabled<br>1'b0: enable<br>1'b1: disable   |
| <b>0x04 I2C Target Address Register</b> |    |           |                     |   |
| [31:13]                                 | RO | 19'b0     | Reserved            | Reserved  |
| [12]                                    | RW | 1'b1      | IC_10BITADDR_MASTER | This bit controls whether the DW_apb_i2c starts its transfers in 7-or 10-bit addressing mode when acting as a master<br>1'b0: 7-bit addressing<br>1'b1: 10-bit addressing   |
| [11]                                    | RW | 1'b0      | SPECIAL             | This bit indicates whether software performs a General Call or START BYTE command<br>1'b0: ignore bit 10 GC_OR_START and use IC_TAR normally<br>1'b1: perform special I2C command as specified in GC_OR_START bit |
| [10]                                    | RW | 1'b0      | GC_OR_START         | If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a General Call or START byte command is to be performed by the DW_apb_i2c<br>1'b0: General Call Address<br>1'b1: START BYTE                      |
| [9:0]                                   | RW | 10'b0x055 | IC_TAR              | This is the target address for any master transaction   |
| <b>0x08 IC_SAR</b>                      |    |           |                     |   |
| [31:10]                                 | RO | 22'b0     | Reserved            | Reserved  |
| [9:0]                                   | RW | 10'b0x055 | IC_SAR              | The IC_SAR holds the slave address when the I2C is operating as a slave. For 7-bit addressing, only IC_SAR[6:0] is used   |
| <b>0x0c IC_HS_MADDR</b>                 |    |           |                     |   |
| [31:3]                                  | RO | 29'b0     | Reserved            | Reserved  |
| [2:0]                                   | RW | 3'b1      | IC_HS_MAR           | This bit field holds the value of the I2C HS mode master code   |
| <b>0x10 IC_DATA_CMD</b>                 |    |           |                     |   |
| [31:11]                                 | RO | 21'b0     | Reserved            | Reserved  |
| [10]                                    | WO | 1'b0      | RESTART             | This bit controls whether a RESTART is issued before the byte is sent or received. This bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1   |
| [9]                                     | WO | 1'b0      | STOP                | This bit controls whether a STOP is issued after the byte is sent or received. This bit is available only if  |



|         |    |                |                |  |
|---------|----|----------------|----------------|--|
|         |    |                |                | IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1   |
|         |    |                |                | This bit controls whether a read or a write is performed   |
| [8]     | WO | 1'b0           | CMD            | 1'b0: Read<br>1'b1: Write  |
| [7:0]   | RW | 8'b0           | DAT            | This register contains the data to be transmitted or received on the I2C bus   |
| 0x14    |    | IC_SS_SCL_HCNT |                |  |
| [31:16] | RO | 16'b0          | Reserved       | Reserved   |
|         |    |                |                | This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed |
| [15:0]  | RW | 16'b0          | IC_SS_SCL_HCNT |  |
| 0x18    |    | IC_SS_SCL_LCNT |                |  |
| [31:16] | RO | 16'b0          | Reserved       | Reserved   |
|         |    |                |                | This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed  |
| [15:0]  | RW | 16'b0          | IC_SS_SCL_LCNT |  |
| 0x1c    |    | IC_FS_SCL_HCNT |                |  |
| [31:16] | RO | 16'b0          | Reserved       | Reserved   |
|         |    |                |                | This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast speed     |
| [15:0]  | RW | 16'b0          | IC_FS_SCL_HCNT |  |
| 0x20    |    | IC_FS_SCL_LCNT |                |  |
| [31:16] | RO | 16'b0          | Reserved       | Reserved   |
|         |    |                |                | This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low-period count for fast speed      |
| [15:0]  | RW | 16'b0          | IC_FS_SCL_LCNT |  |
| 0x24    |    | IC_HS_SCL_HCNT |                |  |
| [31:16] | RO | 16'b0          | Reserved       | Reserved   |
|         |    |                |                | This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high period count for high speed     |
| [15:0]  | RW | 16'b0          | IC_HS_SCL_HCNT |  |
| 0x28    |    | IC_HS_SCL_LCNT |                |  |
| [31:16] | RO | 16'b0          | Reserved       | Reserved   |
|         |    |                |                | This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for high speed      |
| [15:0]  | RW | 16'b0          | IC_HS_SCL_LCNT |  |
| 0x2c    |    | IC_INTR_STAT   |                |  |
| [31:12] | RO | 20'b0          | Reserved       | Reserved   |
| [11]    | RO | 1'b0           | R_GEN_CALL     | Set only when a General Call address is received and it is acknowledged  |
| [10]    | RO | 1'b0           | R_START_DET    | Indicates whether a START or RESTART condition has occurred on the I2C interface   |
| [9]     | RO | 1'b0           | R_STOP_DET     | Indicates whether a STOP condition has occurred on the I2C interface   |

|         |    |                  |             |   |
|---------|----|------------------|-------------|---|
| [8]     | RO | 1'b0             | R_ACTIVITY  | This bit captures DW_apb_i2c activity and stays set until it is cleared   |
| [7]     | RO | 1'b0             | R_RX_DONE   | When the DW_apb_i2c is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte  |
| [6]     | RO | 1'b0             | R_TX_ABRT   | This bit indicates if DW_apb_i2c, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO                                  |
| [5]     | RO | 1'b0             | R_RD_REQ    | This bit is set to 1 when DW_apb_i2c is acting as a slave and another I2C master is attempting to read data from DW_apb_i2c   |
| [4]     | RO | 1'b0             | R_TX_EMPTY  | This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register   |
| [3]     | RO | 1'b0             | R_TX_OVER   | Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register |
| [2]     | RO | 1'b0             | R_RX_FULL   | Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register  |
| [1]     | RO | 1'b0             | R_RX_OVER   | Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device                                       |
| [0]     | RO | 1'b0             | R_RX_UNDER  | Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register  |
| 0x30    |    | IC_INTR_MASK     |             |   |
| [31:12] | RW | 20'b0            | Reserved    | Reserved  |
| [11]    | RW | 1'b1             | R_GEN_CALL  | mask R_GEN_CALL interrupt status bits   |
| [10]    | RW | 1'b0             | R_START_DET | mask R_START_DET interrupt status bits  |
| [9]     | RW | 1'b0             | R_STOP_DET  | mask R_STOP_DET interrupt status bits   |
| [8]     | RW | 1'b0             | R_ACTIVITY  | mask R_ACTIVITY interrupt status bits   |
| [7]     | RW | 1'b1             | R_RX_DONE   | mask R_RX_DONE interrupt status bits  |
| [6]     | RW | 1'b1             | R_TX_ABRT   | mask R_TX_ABRT interrupt status bits  |
| [5]     | RW | 1'b1             | R_RD_REQ    | mask R_RD_REQ interrupt status bits   |
| [4]     | RW | 1'b1             | R_TX_EMPTY  | mask R_TX_EMPTY interrupt status bits   |
| [3]     | RW | 1'b1             | R_TX_OVER   | mask R_TX_OVER interrupt status bits  |
| [2]     | RW | 1'b1             | R_RX_FULL   | mask R_RX_FULL interrupt status bits  |
| [1]     | RW | 1'b1             | R_RX_OVER   | mask R_RX_OVER interrupt status bits  |
| [0]     | RW | 1'b1             | R_RX_UNDER  | mask R_RX_UNDER interrupt status bits   |
| 0x34    |    | IC_RAW_INTR_STAT |             |   |
| [31:12] | RO | 20'b0            | Reserved    | Reserved  |
| [11]    | RO | 1'b0             | GEN_CALL    | Set only when a General Call address is received and it is acknowledged   |
| [10]    | RO | 1'b0             | START_DET   | Indicates whether a START or RESTART condition has occurred on the I2C interface  |
| [9]     | RO | 1'b0             | STOP_DET    | Indicates whether a STOP condition has occurred on the I2C interface  |

|        |    |       |              |   |
|--------|----|-------|--------------|---|
| [8]    | RO | 1'b0  | ACTIVITY     | This bit captures DW_apb_i2c activity and stays set until it is cleared   |
| [7]    | RO | 1'b0  | RX_DONE      | When the DW_apb_i2c is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte  |
| [6]    | RO | 1'b0  | TX_ABRT      | This bit indicates if DW_apb_i2c, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO                                  |
| [5]    | RO | 1'b0  | RD_REQ       | This bit is set to 1 when DW_apb_i2c is acting as a slave and another I2C master is attempting to read data from DW_apb_i2c   |
| [4]    | RO | 1'b0  | TX_EMPTY     | This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register   |
| [3]    | RO | 1'b0  | TX_OVER      | Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register |
| [2]    | RO | 1'b0  | RX_FULL      | Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register  |
| [1]    | RO | 1'b0  | RX_OVER      | Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device                                       |
| [0]    | RO | 1'b0  | RX_UNDER     | Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register  |
| 0x38   |    |       |              | IC_RX_TL  |
| [31:8] | RO | 24'b0 | Reserved     | Reserved  |
| [7:0]  | RW | 8'b0  | RX_TL        | Receive FIFO Threshold Level  |
| 0x3c   |    |       |              | IC_TX_TL  |
| [31:8] | RO | 24'b0 | Reserved     | Reserved  |
| [7:0]  | RW | 8'b0  | TX_TL        | Transmit FIFO Threshold Level   |
| 0x40   |    |       |              | IC_CLR_INTR   |
| [31:1] | RO | 31'b0 | Reserved     | Reserved  |
| [0]    | RO | 1'b0  | CLR_INTR     | Read this register to clear the combined interrupt, all individual interrupts, and the IC_TX_ABRT_SOURCE register   |
| 0x44   |    |       |              | IC_CLR_RX_UNDER   |
| [31:1] | RO | 31'b0 | Reserved     | Reserved  |
| [0]    | RO | 1'b0  | CLR_RX_UNDER | Read this register to clear the RX_UNDER interrupt (bit 0) of the IC_RAW_INTR_STAT register   |
| 0x48   |    |       |              | IC_CLR_RX_OVER  |
| [31:1] | RO | 31'b0 | Reserved     | Reserved  |
| [0]    | RO | 1'b0  | CLR_RX_OVER  | Read this register to clear the RX_OVER interrupt (bit 1) of the IC_RAW_INTR_STAT register  |
| 0x4c   |    |       |              | IC_CLR_TX_OVER  |
| [31:1] | RO | 31'b0 | Reserved     | Reserved  |

|                       |    |       |               |  |
|-----------------------|----|-------|---------------|--|
| [0]                   | RO | 1'b0  | CLR_TX_OVER   | Read this register to clear the TX_OVER interrupt (bit 3) of the IC_RAW_INTR_STAT register   |
| 0x50 IC_CLR_RD_REQ    |    |       |               |  |
| [31:1]                | RO | 31'b0 | Reserved      | Reserved   |
| [0]                   | RO | 1'b0  | CLR_RD_REQ    | Read this register to clear the RD_REQ interrupt (bit 5) of the IC_RAW_INTR_STAT register  |
| 0x54 IC_CLR_TX_ABRT   |    |       |               |  |
| [31:1]                | RO | 31'b0 | Reserved      | Reserved   |
| [0]                   | RO | 1'b0  | CLR_TX_ABRT   | Read this register to clear the TX_ABRT interrupt (bit 6) of the IC_RAW_INTR_STAT register, and the IC_TX_ABRT_SOURCE register   |
| 0x58 IC_CLR_RX_DONE   |    |       |               |  |
| [31:1]                | RO | 31'b0 | Reserved      | Reserved   |
| [0]                   | RO | 1'b0  | CLR_RX_DONE   | Read this register to clear the RX_DONE interrupt (bit 7) of the IC_RAW_INTR_STAT register   |
| 0x5c IC_CLR_ACTIVITY  |    |       |               |  |
| [31:1]                | RO | 31'b0 | Reserved      | Reserved   |
| [0]                   | RO | 1'b0  | CLR_ACTIVITY  | Reading this register clears the ACTIVITY interrupt if the I2C is not active anymore. If the I2C module is still active on the bus, the ACTIVITY interrupt bit continues to be set |
| 0x60 IC_CLR_STOP_DET  |    |       |               |  |
| [31:1]                | RO | 31'b0 | Reserved      | Reserved   |
| [0]                   | RO | 1'b0  | CLR_STOP_DET  | Read this register to clear the STOP_DET interrupt (bit 9) of the IC_RAW_INTR_STAT register  |
| 0x64 IC_CLR_START_DET |    |       |               |  |
| [31:1]                | RO | 31'b0 | Reserved      | Reserved   |
| [0]                   | RO | 1'b0  | CLR_START_DET | Read this register to clear the START_DET interrupt (bit 10) of the IC_RAW_INTR_STAT register  |
| 0x68 IC_CLR_GEN_CALL  |    |       |               |  |
| [31:1]                | RO | 31'b0 | Reserved      | Reserved   |
| [0]                   | RO | 1'b0  | CLR_GEN_CALL  | Read this register to clear the GEN_CALL interrupt (bit 11) of IC_RAW_INTR_STAT register   |
| 0x6c IC_ENABLE        |    |       |               |  |
| [31:2]                | RO | 30'b0 | Reserved      | Reserved   |
| [1]                   | RW | 1'b0  | ABORT         | When set, the controller initiates the transfer abort<br>1'b0: ABORT not initiated or ABORT done<br>1'b1: ABORT operation in progress  |
| [0]                   | RW | 1'b0  | ENABLE        | Controls whether the DW_apb_i2c is enabled<br>1'b0: disable<br>1'b1: enable  |
| 0x70 IC_STATUS        |    |       |               |  |

|         |    |       |                      |  |
|---------|----|-------|----------------------|--|
| [31:7]  | RO | 25'b0 | Reserved             | Reserved   |
| [6]     | RO | 1'b0  | SLV_ACTIVITY         | Slave FSM Activity Status<br>1'b0: in IDLE state<br>1'b1: not in IDLE state  |
| [5]     | RO | 1'b0  | MST_ACTIVITY         | Master FSM Activity Status<br>1'b0: in IDLE state<br>1'b1: not in IDLE state   |
| [4]     | RO | 1'b0  | RFF                  | Receive FIFO Completely Full<br>1'b0: not full<br>1'b1: full   |
| [3]     | RO | 1'b0  | RFNE                 | Receive FIFO Not Empty<br>1'b0: empty<br>1'b1: not empty   |
| [2]     | RO | 1'b1  | TFE                  | Transmit FIFO Completely Empty<br>1'b0: not empty<br>1'b1: empty   |
| [1]     | RO | 1'b1  | TFNF                 | Transmit FIFO Not Full<br>1'b0: full<br>1'b1: not full   |
| [0]     | RO | 1'b0  | ACTIVITY             | I2C Activity Status  |
| 0x74    |    |       |                      | IC_TXFLR   |
| [31:4]  | RO | 28'b0 | Reserved             | Reserved   |
| [3:0]   | RO | 4'b0  | TXFLR                | Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO  |
| 0x78    |    |       |                      | IC_RXFLR   |
| [31:4]  | RO | 28'b0 | Reserved             | Reserved   |
| [3:0]   | RO | 4'b0  | RXFLR                | Receive FIFO Level. Contains the number of valid data entries in the receive FIFO  |
| 0x7c    |    |       |                      | IC_RXFLR   |
| [31:16] | RO | 16'b0 | Reserved             | Reserved   |
| [15:0]  | RW | 16'b1 | IC_SDA_HOLD          | Sets the required SDA hold time in units of ic_clk period  |
| 0x80    |    |       |                      | IC_TX_ABRT_SOURCE  |
| [31:24] | RO | 8'b0  | TX_FLUSH_CNT         | This field preserves the TXFLR value prior to the last TX_ABRT event   |
| [23:17] | RO | 7'b0  | Reserved             | Reserved   |
| [16]    | RO | 1'b0  | ABRT_USER_ABRT       | This is a master-mode-only bit. Master has detected the transfer abort (IC_ENABLE[1])<br>When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of IC_DATA_CMD register |
| [15]    | RO | 1'b0  | ABRT_SLVRD_INTX      | Slave lost the bus while transmitting data to a remote master  |
| [14]    | RO | 1'b0  | ABRT_SLV_ARBLOST     | Slave has received a read command and some data exists in the TX FIFO so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO   |
| [13]    | RO | 1'b0  | ABRT_SLVFLUSH_TXFIFO | Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration  |
| [12]    | RO | 1'b0  | ARB_LOST             | User tries to initiate a Master operation with the Master mode disabled  |
| [11]    | RO | 1'b0  | ABRT_MASTER_DIS      |  |

|        |    |                       |                    |   |
|--------|----|-----------------------|--------------------|---|
| [10]   | RO | 1'b0                  | ABRT_10B_RD_NORSTR | The restart is disabled(IC_RESTART_EN bit (IC_CON[5]) = 0) and the master sends a read command in 10-bit addressing mode  |
| [9]    | RO | 1'b0                  | ABRT_SBYTE_NORSTR  | The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to send a START Byte   |
| [8]    | RO | 1'b0                  | ABRT_HS_NORSTR     | The restart is disabled(IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to use the master to transfer data in High Speed mode   |
| [7]    | RO | 1'b0                  | ABRT_SBYTE_ACKDET  | Master has sent a START Byte and the START Byte was acknowledged (wrong behavior)   |
| [6]    | RO | 1'b0                  | ABRT_HS_ACKDET     | Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior)   |
| [5]    | RO | 1'b0                  | ABRT_GCALL_READ    | DW_apb_i2c in master mode sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1)  |
| [4]    | RO | 1'b0                  | ABRT_GCALL_NOACK   | DW_apb_i2c in master mode sent a General Call and no slave on the bus acknowledged the General Call   |
| [3]    | RO | 1'b0                  | ABRT_TXDATA_NOACK  | This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s) |
| [2]    | RO | 1'b0                  | ABRT_10ADDR2_NOACK | Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave  |
| [1]    | RO | 1'b0                  | ABRT_10ADDR1_NOACK | Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave  |
| [0]    | RO | 1'b0                  | ABRT_7B_ADDR_NOACK | Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave   |
| 0x84   |    | IC_SLV_DATA_NACK_ONLY |                    |   |
| [31:1] | RO | 31'b0                 | Reserved           | Reserved  |
| [0]    | RW | 1'b0                  | NACK               | Generate NACK<br>1'b0: generate NACK after data byte received<br>1'b1: generate NACK/ACK normally   |
| 0x88   |    | IC_DMA_CR             |                    |   |
| [31:2] | RO | 30'b0                 | Reserved           | Reserved  |
| [1]    | RW | 1'b0                  | TDMAE              | Transmit DMA Enable<br>1'b0: disable<br>1'b1: enable  |
| [0]    | RW | 1'b0                  | RDMAE              | Receive DMA Enable<br>1'b0: disable<br>1'b1: enable   |
| 0x8c   |    | IC_DMA_TDLR           |                    |   |
| [31:3] | RO | 29'b0                 | Reserved           | Reserved  |
| [2:0]  | RW | 3'b0                  | DMATDL             | Transmit Data Level   |
| 0x90   |    | IC_DMA_RDLR           |                    |   |

|         |    |                     |                         |   |
|---------|----|---------------------|-------------------------|---|
| [31:3]  | RO | 29'b0               | Reserved                | Reserved  |
| [2:0]   | RW | 3'b0                | DMARDL                  | Receive Data Level  |
| 0x94    |    | IC_SDA_SETUP        |                         |   |
| [31:8]  | RO | 24'b0               | Reserved                | Reserved  |
| [7:0]   | RW | 8'b0x64             | SDA_SETUP               | SDA Setup   |
| 0x98    |    | IC_ACK_GENERAL_CALL |                         |   |
| [31:1]  | RO | 31'b0               | Reserved                | Reserved  |
| [0]     | RW | 1'b1                | ACK_GEN_CALL            | ACK General Call  |
| 0x9c    |    | IC_ENABLE_STATUS    |                         |   |
| [31:3]  | RO | 31'b0               | Reserved                | Reserved  |
| [2]     | RO | 1'b0                | SLV_RX_DATA_LOST        | Slave Received Data Lost  |
| [1]     | RO | 1'b0                | SLV_DISABLED_WHILE_BUSY | Slave Disabled While Busy (Transmit, Receive)   |
|         |    |                     |                         | ic_en Status  |
|         |    |                     |                         | 1'b0: DW_apb_i2c is deemed completely inactive  |
|         |    |                     |                         | 1'b1: DW_apb_i2c is deemed to be in an enabled state  |
| 0xa0    |    | IC_FS_SPKLEN        |                         |   |
| [31:8]  | RO | 24'b0               | Reserved                | Reserved  |
| [7:0]   | RW | 8'b0xff             | IC_FS_SPKLEN            | This register must be set before any I2C bus transaction can take place to ensure stable operation                              |
| 0xa4    |    | IC_HS_SPKLEN        |                         |   |
| [31:8]  | RO | 24'b0               | Reserved                | Reserved  |
| [7:0]   | RW | 8'b0xff             | IC_HS_SPKLEN            | This register must be set before any I2C bus transaction can take place to ensure stable operation                              |
| 0xf4    |    | IC_COMP_PARAM_1     |                         |   |
| [31:24] | RO | 8'b0                | Reserved                | Reserved  |
|         |    |                     |                         | The value of this register is derived from the IC_TX_BUFFER_DEPTH coreConsultant parameter                                      |
|         |    |                     |                         | 8'b0x00: Reserved   |
|         |    |                     |                         | 8'b0x01: 2  |
|         |    |                     |                         | 8'b0x02: 3  |
|         |    |                     |                         | ...   |
|         |    |                     |                         | 8'b0xff: 256  |
|         |    |                     |                         | The value of this register is derived from the IC_RX_BUFFER_DEPTH coreConsultant parameter. For a description of this parameter |
|         |    |                     |                         | 8'b0x00: Reserved   |
|         |    |                     |                         | 8'b0x01: 2  |
|         |    |                     |                         | 8'b0x02: 3  |
|         |    |                     |                         | ...   |
|         |    |                     |                         | 8'b0xff: 256  |
|         |    |                     |                         | The value of this register is derived from the IC_ADD_ENCODED_PARAMS coreConsultant parameter.                                  |
|         |    |                     |                         | 1'b0: False   |
| [7]     | RO | 1'b0                | ADD_ENCODED_PARAMS      |   |

1'b1: True

|       |    |      |                 |  |
|-------|----|------|-----------------|--|
| [6]   | RO | 1'b0 | HAS_DMA         | The value of this register is derived from the IC_HAS_DMA coreConsultant parameter<br>1'b0: False<br>1'b1: True  |
| [5]   | RO | 1'b0 | INTR_IO         | The value of this register is derived from the IC_INTR_IO coreConsultant parameter<br>1'b0: Individual<br>1'b1: Combined                                       |
| [4]   | RO | 1'b0 | HC_COUNT_VALUES | The value of this register is derived from the IC_HC_COUNT_VALUES coreConsultant parameter<br>1'b0: False<br>1'b1: True  |
| [3:2] | RO | 2'b0 | MAX_SPEED_MODE  | The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter<br>2'b00: Reserved<br>2'b01: Standard<br>2'b10: Fast<br>2'b11: High  |
| [1:0] | RO | 2'b0 | APB_DATA_WIDTH  | The value of this register is derived from the APB_DATA_WIDTH coreConsultant parameter<br>2'b00: 8 bits<br>2'b01: 16 bits<br>2'b10: 32 bits<br>2'b11: Reserved |

#### 0xf8 IC\_COMP\_VERSION

|        |    |       |                 |   |
|--------|----|-------|-----------------|---|
| [31:0] | RO | 32'b0 | IC_COMP_VERSION | Specific values for this register are described in the Releases Table in the AMBA 2 release notes |
|--------|----|-------|-----------------|---|

#### 0xfc IC\_COMP\_TYPE

|        |    |       |              |  |
|--------|----|-------|--------------|--|
| [31:0] | RO | 32'b0 | IC_COMP_TYPE | This assigned unique hex value is constant and is derived from the two ASCII letters "DW" followed by a 16-bit unsigned number |
|--------|----|-------|--------------|--|

## 4.9 I2S

I2S wrapper contains one I2S master and one I2S slave. They are logically exclusive. Only one block is alive at a time. The operation mode for master mode and slave mode is controlled by PERI\_MASTER\_SELECT Register in COM block.

| bit | Reset value | Definition                   |
|-----|-------------|------------------------------|
| 3   | 0           | I2S1 is master mode when set |
| 2   | 0           | I2S0 is master mode when set |

**Table 11: PERI\_MASTER\_SELECT Register bit definition  
(base address = 0x4002\_302C)**

I2S registers are listed below:



Base address: 4000\_9000

|         |    |       |          |  |  |
|---------|----|-------|----------|--|--|
| 0x00    |    |       |          |  | IER  |
| [31:1]  | RO | 31'b0 | Reserved | Reserved   | DW_apb_i2s enable  |
| [0]     | RW | 1'b0  | IEN      | 1'b0: disable<br>1'b1: enable  |  |
| 0x04    |    |       |          |  | IRER   |
| [31:1]  | RO | 31'b0 | Reserved | Reserved   | Receiver block enable  |
| [0]     | RW | 1'b0  | RXEN     | 1'b0: disable<br>1'b1: enable  |  |
| 0x08    |    |       |          |  | ITER   |
| [31:1]  | RO | 31'b0 | Reserved | Reserved   | Transmitter block enable   |
| [0]     | RW | 1'b0  | TXEN     | 1'b0: disable<br>1'b1: enable  |  |
| 0x0c    |    |       |          |  | CER  |
| [31:1]  | RO | 31'b0 | Reserved | Reserved   | Clock generation enable/disable  |
| [0]     | RW | 1'b0  | CLKEN    | 1'b0: disable<br>1'b1: enable  |  |
| 0x10    |    |       |          |  | CCR  |
| [31:5]  | RO | 27'b0 | Reserved | Reserved   | These bits are used to program the number of<br>sclk cycles for which the world select<br>line(ws_out) stays in the left or right sample<br>mode |
| [4:3]   | RW | 2'b00 | WSS      | 2'b00: 16 clock cycles<br>2'b01: 24 clock cycles<br>2'b10: 32 clock cycles   |  |
| [2:0]   | RW | 3'b0  | SCLKG    | These bits are used to program the gating of sclk<br>3'b000: No clock gating<br>3'b001: Gate after 12 clock cycles<br>3'b010: Gate after 16 clock cycles<br>3'b011: Gate after 20 clock cycles<br>3'b100: Gate after 24 clock cycles |  |
| 0x14    |    |       |          |  | RXFFR  |
| [31:1]  | RO | 31'b0 | Reserved | Reserved   | Receiver FIFO Reset;Receiver Block must be<br>disabled prior to writing this bit   |
| [0]     | WO | 1'b0  | RXFFR    |  |  |
| 0x18    |    |       |          |  | TXFFR  |
| [31:1]  | RO | 31'b0 | Reserved | Reserved   | Transmitter FIFO Reset;Transmitter Block must<br>be disabled prior to writing this bit   |
| [0]     | WO | 1'b0  | TXFFR    |  |  |
| 0x20    |    |       |          |  | LRBR0  |
| [31:16] | RO | 16'b0 | Reserved | Reserved   | The left stereo data received serially from the<br>receive channel input is read through this<br>register  |
| [15:0]  | RO | 16'b0 | LRBR0    |  |  |
| 0x20    |    |       |          |  | LTHR0  |

|            |    |        |          |   |
|------------|----|--------|----------|---|
| [31:16]    | RO | 16'b0  | Reserved | Reserved  |
| [15:0]     | WO | 16'b0  | LTHRO    | The left stereo to be transmitted serially through the transmit channel output is written through this register   |
| 0x24 RRBRO |    |        |          |   |
| [31:16]    | RO | 16'b0  | Reserved | Reserved  |
| [15:0]     | RO | 16'b0  | RRBRO    | The right stereo data received serially from the receive channel input is read through this register  |
| 0x24 RTHRO |    |        |          |   |
| [31:16]    | RO | 16'b0  | Reserved | Reserved  |
| [15:0]     | WO | 16'b0  | RTHRO    | The right stereo to be transmitted serially through the transmit channel output is written through this register  |
| 0x28 RERO  |    |        |          |   |
| [31:1]     | RO | 31'b0  | Reserved | Reserved  |
| [0]        | RW | 1'b1   | RXCHENO  | Receive channel enable<br>1'b0: disable<br>1'b1: enable   |
| 0x2c TERO  |    |        |          |   |
| [31:1]     | RO | 31'b0  | Reserved | Reserved  |
| [0]        | RW | 1'b1   | TXCHENO  | Transmit channel enable<br>1'b0: disable<br>1'b1: enable  |
| 0x30 RCRO  |    |        |          |   |
| [31:3]     | RO | 29'b0  | Reserved | Reserved  |
|            |    |        |          | These bits are used to program the desired data resolution of the receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBR0(or RRBE0) register |
| [2:0]      | RW | 3'b010 | WLEN     | 3'b000: Ignore word length<br>3'b001: 12 bit resolution<br>3'b010: 16 bit resolution<br>3'b011: 20 bit resolution<br>3'b100: 24 bit resolution<br>3'b101: 32 bit resolution                 |
| 0x34 TCRO  |    |        |          |   |
| [31:3]     | RO | 29'b0  | Reserved | Reserved  |
|            |    |        |          | These bits are used to program the data resolution of the transmitter and ensure the MSB of the data is transmitted first   |
| [2:0]      | RW | 3'b010 | WLEN     | 3'b000: Ignore word length<br>3'b001: 12 bit resolution<br>3'b010: 16 bit resolution<br>3'b011: 20 bit resolution<br>3'b100: 24 bit resolution<br>3'b101: 32 bit resolution                 |
| 0x38 ISRO  |    |        |          |   |
| [31:6]     | RO | 26'b0  | Reserved | Reserved  |

|        |    |        |          |   |
|--------|----|--------|----------|---|
| [5]    | RO | 1'b0   | TXFO     | Status of Data Overrun interrupt for the TX channel<br>1'b0: TX FIFO write valid<br>1'b1: TX FIFO write overrun                       |
| [4]    | RO | 1'b1   | TXFE     | Status of Transmit Empty Trigger interrupt<br>1'b0: trigger level not reached<br>1'b1: trigger level reached                          |
| [3:2]  | RO | 2'b0   | Reserved | Reserved  |
| [1]    | RO | 1'b0   | RXFO     | Status of Data Overrun interrupt for the RX channel<br>1'b0: RX FIFO write valid<br>1'b1: RX FIFO write overrun                       |
| [0]    | RO | 1'b0   | RXDA     | Status of Receive Data Available interrupt<br>1'b0: trigger level not reached<br>1'b1: trigger level reached                          |
| 0x3c   |    | IMR0   |          |   |
| [31:6] | RO | 26'b0  | Reserved | Reserved  |
| [5]    | RW | 1'b1   | TXFOM    | Masks TX FIFO Overrun interrupt<br>1'b0: unmask interrupt<br>1'b1: mask interrupt   |
| [4]    | RW | 1'b1   | TXFEM    | Masks TX FIFO Empty interrupt<br>1'b0: unmask interrupt<br>1'b1: mask interrupt   |
| [3:2]  | RO | 2'b0   | Reserved | Reserved  |
| [1]    | RW | 1'b1   | RXFOM    | Masks RX FIFO Overrun interrupt<br>1'b0: unmask interrupt<br>1'b1: mask interrupt   |
| [0]    | RW | 1'b1   | RXDAM    | Masks RX FIFO Data Available interrupt<br>1'b0: unmask interrupt<br>1'b1: mask interrupt  |
| 0x40   |    | ROR0   |          |   |
| [31:1] | RO | 31'b0  | Reserved | Reserved  |
| [0]    | RO | 1'b1   | RXCHO    | Read this bit to clear the RX FIFO Data Overrun interrupt<br>1'b0: RX FIFO write valid<br>1'b1: RX FIFO write overrun                 |
| 0x44   |    | TOR0   |          |   |
| [31:1] | RO | 31'b0  | Reserved | Reserved  |
| [0]    | RO | 1'b1   | TXCHO    | Read this bit to clear the TX FIFO Data Overrun interrupt<br>1'b0: TX FIFO write valid<br>1'b1: TX FIFO write overrun                 |
| 0x48   |    | RFCR0  |          |   |
| [31:4] | RO | 29'b0  | Reserved | Reserved  |
| [3:0]  | RW | 3'b011 | RXCHDT   | These bits program the trigger level in the RX FIFO at which the Received Data Available interrupt is generated                       |
| 0x4c   |    | TFCR0  |          |   |
| [31:4] | RO | 29'b0  | Reserved | Reserved  |
| [3:0]  | RW | 3'b011 | TXCHET   | Transmit Channel Empty Trigger; These bits program the trigger level in the TX FIFO at which the Empty Threshold Reached interrupt is |

generated

|         |    |       |                   |   |
|---------|----|-------|-------------------|---|
| 0x50    |    |       |                   | RFF0  |
| [31:1]  | RO | 31'b0 | Reserved          | Reserved  |
| [0]     | WO | 1'b0  | RXCHFR            | Receive Channel FIFO Reset;Writing a 1 to this register flushes an individual RX FIFO,Rx channel or block must be disabled prior to writing this bit                  |
| 0x54    |    |       |                   | TFF0  |
| [31:1]  | RO | 31'b0 | Reserved          | Reserved  |
| [0]     | WO | 1'b0  | TXCHFR            | Transmit Channel FIFO Reset;Writing a 1 to this register flushes channel's TX FIFO,Tx channel or block must be disabled prior to writing this bit                     |
| 0x1c0   |    |       |                   | RXDMA   |
| [31:0]  | RO | 32'b0 | RXDMA             | Receiver Block DMA Register.Used to cycle repeatedly through the enabled receive channels(from lowest numbered to highest),reading stereo data pairs                  |
| 0x1c4   |    |       |                   | RRXDMA  |
| [31:1]  | RO | 31'b0 | Reserved          | Reserved  |
| [0]     | WO | 1'b0  | RRXDMA            | Reset Receiver Block DMA Register.Writing a 1 to this self-clearing register reset the RXDMA register mid-cycle to point to the lowest enabled Receive channel        |
| 0x1c8   |    |       |                   | TXDMA   |
| [31:0]  | RO | 32'b0 | TXDMA             | Transmitter Block DMA Register.Used to cycle repeatedly through the enabled receive channels(from lowest numbered to highest),reading stereo data pairs               |
| 0x1cc   |    |       |                   | RTXDMA  |
| [31:1]  | RO | 31'b0 | Reserved          | Reserved  |
| [0]     | WO | 1'b0  | RTXDMA            | Reset Transmitter Block DMA Register.Writing a 1 to this self-clearing register reset the TXDMA register mid-cycle to point to the lowest enabled Receive channel     |
| 0x1f0   |    |       |                   | I2S_COMP_PARAM_2  |
| [31:13] | RO | 19'b0 | Reserved          | Reserved  |
| [12:10] | RO | 3'b0  | I2S_RX_WORDSIZE_3 | 3'b000: 12 bit resolution<br>3'b001: 16 bit resolution<br>3'b010: 20 bit resolution<br>3'b011: 24 bit resolution<br>3'b100: 32 bit resolution<br>3'b101~111: Reserved |
| [9:7]   | RO | 3'b0  | I2S_RX_WORDSIZE_2 | 3'b000: 12 bit resolution<br>3'b001: 16 bit resolution<br>3'b010: 20 bit resolution<br>3'b011: 24 bit resolution<br>3'b100: 32 bit resolution<br>3'b101~111: Reserved |
| [6]     | RO | 1'b0  | Reserved          | Reserved  |

|                        |    |      |                       |   |
|------------------------|----|------|-----------------------|---|
| [5:3]                  | RO | 3'b0 | I2S_RX_WORDSIZE_1     | 3'b000: 12 bit resolution<br>3'b001: 16 bit resolution<br>3'b010: 20 bit resolution<br>3'b011: 24 bit resolution<br>3'b100: 32 bit resolution<br>3'b101~111: Reserved |
| [2:0]                  | RO | 3'b0 | I2S_RX_WORDSIZE_0     | 3'b000: 12 bit resolution<br>3'b001: 16 bit resolution<br>3'b010: 20 bit resolution<br>3'b011: 24 bit resolution<br>3'b100: 32 bit resolution<br>3'b101~111: Reserved |
| 0x1f4 I2S_COMP_PARAM_1 |    |      |                       |   |
| [31:28]                | RO | 4'b0 | Reserved              | Reserved  |
| [27:25]                | RO | 3'b0 | I2S_TX_WORDSIZE_3     | 3'b000: 12 bit resolution<br>3'b001: 16 bit resolution<br>3'b010: 20 bit resolution<br>3'b011: 24 bit resolution<br>3'b100: 32 bit resolution<br>3'b101~111: Reserved |
| [24:22]                | RO | 3'b0 | I2S_TX_WORDSIZE_2     | 3'b000: 12 bit resolution<br>3'b001: 16 bit resolution<br>3'b010: 20 bit resolution<br>3'b011: 24 bit resolution<br>3'b100: 32 bit resolution<br>3'b101~111: Reserved |
| [21:19]                | RO | 3'b0 | I2S_TX_WORDSIZE_1     | 3'b000: 12 bit resolution<br>3'b001: 16 bit resolution<br>3'b010: 20 bit resolution<br>3'b011: 24 bit resolution<br>3'b100: 32 bit resolution<br>3'b101~111: Reserved |
| [18:16]                | RO | 3'b0 | I2S_TX_WORDSIZE_0     | 3'b000: 12 bit resolution<br>3'b001: 16 bit resolution<br>3'b010: 20 bit resolution<br>3'b011: 24 bit resolution<br>3'b100: 32 bit resolution<br>3'b101~111: Reserved |
| [15:11]                | RO | 5'b0 | Reserved              | Reserved  |
| [10:9]                 | RO | 2'b0 | I2S_TX_CHANNELS       | 2'b00: 1 channel<br>2'b01: 2 channels<br>2'b10: 3 channels<br>2'b11: 4 channels   |
| [8:7]                  | RO | 2'b0 | I2S_RX_CHANNELS       | 2'b00: 1 channel<br>2'b01: 2 channels<br>2'b10: 3 channels<br>2'b11: 4 channels   |
| [6]                    | RO | 1'b0 | I2S_RECEIVER_BLOCK    | 1'b0: FALSE<br>1'b1: TRUE   |
| [5]                    | RO | 1'b0 | I2S_TRANSIMITER_BLOCK | 1'b0: FALSE<br>1'b1: TRUE   |
| [4]                    | RO | 1'b0 | I2S_MODE_EN           | 1'b0: FALSE<br>1'b1: TRUE   |

|       |    |      |                       |   |
|-------|----|------|-----------------------|---|
| [3:2] | RO | 2'b0 | I2S_FIFO_DEPTH_GLONAL | 2'b00: 2<br>2'b01: 4<br>2'b10: 8<br>2'b11: 16         |
| [1:0] | RO | 2'b0 | APB_DATA_WIDTH        | 2'b00: 8<br>2'b01: 16<br>2'b10: 32<br>2'b11: Reserved |

0x1f8 I2S\_COMP\_VERSION

|         |    |       |                  |   |
|---------|----|-------|------------------|---|
| [31:28] | RO | 32'b0 | I2S_COMP_VERSION | Specific values for this register are described in the Releases Table in the AMBA 2 release notes |
|---------|----|-------|------------------|---|

## 4.10 UART (UART)

The Universal Asynchronous Receiver/Transmitter offers fast, full-duplex, asynchronous serial communication with built-in flow control (CTS, RTS) support in HW up to 1Mbps baud. Parity checking and generation for the 9th data bit are supported.

The GPIOs used for each UART interface line can be chosen from any GPIO on the device and are independently configurable. This enables great flexibility in device pin out and enables efficient use of board space and signal routing.

UART registers are listed below:

Base address: 4000\_4000

| OFFSET | TYPE | RESET | NAME                           | DESCRIPTION   |
|--------|------|-------|--------------------------------|---|
| 0x00   |      |       | RBR(Receive Buffer Register)   | LCR[7] bit = 0  |
| [31:8] | RO   | 24'b0 | Reserved                       | Reserved  |
| [7:0]  | RO   | 8'b0  | Receive Buffer Register        | LSR[0] bit = 1, The data in this register is valid  |
| 0x00   |      |       | THR(Transmit Holding Register) | LCR[7] bit = 0  |
| [31:8] | WO   | 24'b0 | Reserved                       | Reserved  |
| [7:0]  | WO   | 8'b0  | Transmit Holding Register      | LSR[5] bit = 1, The data should only be written to the THR  |
| 0x00   |      |       | DLL(Divisor Latch Low)         | 1. When UART_16550 == YES, Then LCR[7] bit = 1<br>2. When UART_16550 == NO, Then LCR[7] bit = 1, USR[0] = 0 |
| [31:8] | RO   | 24'b0 | Reserved                       | Reserved  |
| [7:0]  | RW   | 8'b0  | Divisor Latch (low)            | baud rate = (serial clock freq) / (16 * divisor)  |
| 0x04   |      |       | DLH(Divisor Latch High)        | 1. When UART_16550 == YES, Then LCR[7] bit = 1<br>2. When UART_16550 == NO, Then LCR[7] bit = 1, USR[0] = 0 |
| [31:8] | RO   | 24'b0 | Reserved                       | Reserved  |

|        |    |                                  |                      |   |
|--------|----|----------------------------------|----------------------|---|
| [7:0]  | RW | 8'b0                             | Divisor Latch (high) | baud rate = (serial clock freq) / (16 * divisor)  |
| 0x04   |    | IER(Interrupt Enable Register)   |                      | LCR[7] bit = 0  |
| [31:8] | RO | 24'b0                            | Reserved             | Reserved  |
| [7]    | RW | 1'b0                             | PTIME                | This is used to enable/disable the generation of THRE Interrupt<br>1'b0: disable<br>1'b1: enable  |
| [6:4]  | RO | 3'b0                             | Reserved             | Reserved  |
| [3]    | RW | 1'b0                             | EDSSI                | This is used to enable/disable the generation of Modem Status Interrupt<br>1'b0: disable<br>1'b1: enable  |
| [2]    | RW | 1'b0                             | ELSI                 | This is used to enable/disable the generation of Receiver Line Status Interrupt<br>1'b0: disable<br>1'b1: enable  |
| [1]    | RW | 1'b0                             | ETBEI                | This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt<br>1'b0: disable<br>1'b1: enable  |
| [0]    | RW | 1'b0                             | ERBFI                | This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled)<br>1'b0: disable<br>1'b1: enable   |
| 0x08   |    | IIR(Interrupt Identity Register) |                      |   |
| [31:8] | RO | 24'b0                            | Reserved             | Reserved  |
| [7:6]  | RO | 2'b0                             | FIFOSE               | This is used to indicate whether the FIFOs are enabled or disabled<br>2'b00: disable<br>2'b11: enable   |
| [5:4]  | RO | 2'b0                             | Reserved             | Reserved  |
| [3:0]  | RO | 4'b0001                          | IID                  | This is used to indicates the highest priority pending interrupt which can be one of the following types<br>4'b0000: modem status<br>4'b0001: no interrupt pending<br>4'b0010: THR empty<br>4'b0100: received data available<br>4'b0110: receiver line status<br>4'b0111: busy detect<br>4'b1100: character timeout |
| 0x08   |    | FCR(FIFO Control Register)       |                      | FIFO_MODE != NONE   |
| [31:8] | RO | 24'b0                            | Reserved             | Reserved  |

|                                 |    |       |          |  |
|---------------------------------|----|-------|----------|--|
| [7:6]                           | WO | 2'b0  | RT       | <p>This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated, The following trigger levels are supported:</p> <p>2'b00: 1 character in the FIFO</p> <p>2'b01: FIFO ¼ full</p> <p>2'b10: FIFO ½ full</p> <p>2'b11: FIFO 2 less than full</p> <p>This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active, The following trigger levels are supported:</p> <p>2'b00: FIFO empty</p> <p>2'b01: 2 characters in the FIFO</p> <p>2'b10: FIFO ¼ full</p> <p>2'b11: FIFO ½ full</p> <p>This determines the DMA signalling mode</p> <p>1'b0: mode 0</p> <p>1'b1: mode 1</p> |
| [5:4]                           | WO | 2'b0  | TET      | <p>This resets the control portion of the transmit FIFO and treats the FIFO as empty</p>   |
| [3]                             | WO | 1'b0  | DMAM     | <p>This resets the control portion of the receive FIFO and treats the FIFO as empty</p>  |
| [2]                             | WO | 1'b0  | XFIFOR   | <p>This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs</p> <p>1'b0: disable</p> <p>1'b1: enable</p>   |
| [1]                             | WO | 1'b0  | RFIFOR   |  |
| [0]                             | WO | 1'b0  | FIFOE    |  |
| 0x0C LCR(Line Control Register) |    |       |          |  |
| [31:8]                          | RO | 24'b0 | Reserved | Reserved   |
| [7]                             | RW | 1'b0  | DLAB     | <p>USR[0]=0, the bit is writeable; This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART</p> <p>1'b0: disable</p> <p>1'b1: enable</p>  |
| [6]                             | RW | 1'b0  | Break    | <p>This is used to cause a break condition to be transmitted to the receiving device</p>   |
| [5]                             | RO | 1'b0  | Reserved | Reserved   |
| [4]                             | RW | 1'b0  | EPS      | <p>USR[0]=0, the bit is writeable; This is used to select between even and odd parity, when parity is enabled (PEN set to one)</p> <p>1'b0: an odd number of logic 1s is transmitted or checked</p> <p>1'b1: an even number of logic 1s is transmitted or checked</p>  |



|                                  |    |       |          |   |
|----------------------------------|----|-------|----------|---|
| [3]                              | RW | 1'b0  | PEN      | <p>USR[0]=0,the bit is writeable;enable and disable parity generation and detection in transmitted and received serial character respectively</p> <p>1'b0: disable</p> <p>1'b1: enable</p> <p>USR[0]=0,the bit is writeable;select the number of stop bits per character that the peripheral transmits and receives</p> <p>1'b0: 1 stop bit</p> <p>1'b1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit</p> <p>USR[0]=0,the bit is writeable;This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows:</p> <p>2'b00: 5 bits</p> <p>2'b01: 6 bits</p> <p>2'b10: 7 bits</p> <p>2'b11: 8 bits</p> |
| [2]                              | RW | 1'b0  | STOP     |   |
| [1:0]                            | RW | 2'b0  | DLS      |   |
| 0x10 MCR(Modem Control Register) |    |       |          |   |
| [31:7]                           | RO | 25'b0 | Reserved | Reserved  |
| [6]                              | RW | 1'b0  | SIRE     | <p>SIR_MODE == Enabled,the bit is writeable;enable/disable the IrDA SIR Mode</p> <p>1'b0: disable</p> <p>1'b1: enable</p> <p>AFCE_MODE == Enabled,the bit is writeable;enable/disable the Auto Flow Control</p> <p>1'b0: disable</p> <p>1'b1: enable</p>  |
| [5]                              | RW | 1'b0  | AFCE     |   |
| [4]                              | RW | 1'b0  | LoopBack | <p>This is used to put the UART into a diagnostic mode for test purposes</p> <p>This is used to directly control the user-designated Output2 (out2_n) output</p> <p>1'b0: de-asserted (logic 1)</p> <p>1'b1: asserted (logic 0)</p>   |
| [3]                              | RW | 1'b0  | OUT2     |   |
| [2]                              | RW | 1'b0  | OUT1     | <p>This is used to directly control the user-designated Output1 (out1_n) output</p> <p>1'b0: de-asserted (logic 1)</p> <p>1'b1: asserted (logic 0)</p>  |
| [1]                              | RW | 1'b0  | RTS      | <p>Request to Send. This is used to directly control the Request to Send (rts_n) output</p> <p>1'b0: de-asserted (logic 1)</p> <p>1'b1: asserted (logic 0)</p>  |

|        |    |                            |          |  |
|--------|----|----------------------------|----------|--|
| [0]    | RW | 1'b0                       | DTR      | This is used to directly control the Data Terminal Ready (dtr_n) output<br>1'b0: de-asserted (logic 1)<br>1'b1: asserted (logic 0)   |
| 0x14   |    | LSR(Line Status Register)  |          |  |
| [31:8] | RO | 24'b0                      | Reserved | Reserved   |
| [7]    | RO | 1'b0                       | RFE      | FIFO_MODE != NONE and FCR[0] = 1,the bit is relevant;This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO<br>1'b0: no error<br>1'b1: error   |
| [6]    | RO | 1'b1                       | TEMT     | Transmitter Empty bit;<br>FIFO_MODE != NONE and FCR[0] = 1,this bit is set whenever the Transmitter Shift Register and the FIFO are both empty<br>FIFO_MODE == NONE and FCR[0] = 0,this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty |
| [5]    | RO | 1'b1                       | THRE     | Transmit Holding Register Empty bit  |
| [4]    | RO | 1'b0                       | BI       | This is used to indicate the detection of a break sequence on the serial input data.   |
| [3]    | RO | 1'b0                       | FE       | This is used to indicate the occurrence of a framing error in the receiver<br>1'b0: no framing error<br>1'b1: framing error  |
| [2]    | RO | 1'b0                       | PE       | This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set<br>1'b0: no parity error<br>1'b1: parity error  |
| [1]    | RO | 1'b0                       | OE       | This is used to indicate the occurrence of an overrun error<br>1'b0: no overrun error<br>1'b1: overrun error   |
| [0]    | RO | 1'b0                       | DR       | This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO<br>1'b0: no data ready<br>1'b1: data ready  |
| 0x18   |    | MSR(Modem Status Register) |          |  |
| [31:8] | RO | 24'b0                      | Reserved | Reserved   |
| [7]    | RO | 1'b0                       | DCD      | This is used to indicate the current state of the modem control line dcd_n<br>1'b0: de-asserted (logic 1)<br>1'b1: asserted (logic 0)  |

|     |    |      |      |  |
|-----|----|------|------|--|
| [6] | RO | 1'b0 | RI   | This is used to indicate the current state of the modem control line ri_n<br>1'b0: de-asserted (logic 1)<br>1'b1: asserted (logic 0)   |
| [5] | RO | 1'b0 | DSR  | This is used to indicate the current state of the modem control line dsr_n<br>1'b0: de-asserted (logic 1)<br>1'b1: asserted (logic 0)  |
| [4] | RO | 1'b0 | CTS  | This is used to indicate the current state of the modem control line cts_n<br>1'b0: de-asserted (logic 1)<br>1'b1: asserted (logic 0)  |
| [3] | RO | 1'b0 | DDCD | This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read<br>1'b0: no change on dcd_n since last read of MSR<br>1'b1: change on dcd_n since last read of MSR       |
| [2] | RO | 1'b0 | TERI | This is used to indicate that a change on the input ri_n has occurred since the last time the MSR was read<br>1'b0: no change on ri_n since last read of MSR<br>1'b1: change on ri_n since last read of MSR          |
| [1] | RO | 1'b0 | DDSR | This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read<br>1'b0: no change on dsr_n since last read of MSR<br>1'b1: change on dsr_n since last read of MSR       |
| [0] | RO | 1'b0 | DCTS | This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read<br>1'b0: no change on ctsdsr_n since last read of MSR<br>1'b1: change on ctsdsr_n since last read of MSR |

|        |    |   |                     |  |
|--------|----|---|---------------------|--|
| 0x1C   |    | SCR(Scratchpad Register)                    |                     |  |
| [31:8] | RO | 24'b0                                       | Reserved            | Reserved   |
| [7:0]  | RW | 8'b0  | Scratchpad Register | This register is for programmers to use as a temporary storage space   |
| 0x20   |    | LPDLL(Low Power Divisor Latch Low Register) |                     | SIR_LP_RX == Yes   |
| [31:8] | RO | 24'b0                                       | Reserved            | Reserved   |
| [7:0]  | RW | 8'b0  | LPDLL               | This register makes up the lower 8-bits of a 16-bit, this register that contains the baud rate divisor for the |

## UART

|           |    |       |                                  |  |   |
|-----------|----|-------|----------------------------------|--|---|
| 0x24      |    |       |                                  | LPDLH(Low Power Divisor Latch High Register) | SIR_LP_RX == Yes  |
| [31:8]    | RO | 24'b0 | Reserved                         | Reserved                                     | This register makes up the upper 8-bits of a 16-bit, this register that contains the baud rate divisor for the UART   |
| [7:0]     | RW | 8'b0  | LPDLH                            |  |   |
| 0x30~0x6c |    |       |                                  | SRBR(Shadow Receive Buffer Register)         | SHADOW == YES and LCR[7] bit = 0  |
| [31:8]    | RO | 24'b0 | Reserved                         | Reserved                                     | This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master   |
| [7:0]     | RO | 8'b0  | Shadow Receive Buffer Register   |  |   |
| 0x30~0x6c |    |       |                                  | STHR(Shadow Transmit Holding Register)       | SHADOW == YES and LCR[7] bit = 0  |
| [31:8]    | RO | 24'b0 | Reserved                         | Reserved                                     | This is a shadow register for the THR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master   |
| [7:0]     | WO | 8'b0  | Shadow Transmit Holding Register |  |   |
| 0x70      |    |       |                                  | FAR(FIFO Access Register)                    |   |
| [31:1]    | RO | 31'b0 | Reserved                         | Reserved                                     | Writes have no effect when FIFO_ACCESS == No, always readable, This register is used to enable a FIFO access mode for testing<br>1'b0: disable<br>1'b1: enable  |
| [0]       | RW | 1'b0  | FIFO Access Register             |  |   |
| 0x74      |    |       |                                  | TFR(Transmit FIFO Read)                      | FIFO_ACCESS == YES  |
| [31:8]    | RO | 24'b0 | Reserved                         | Reserved                                     | FAR[0] = 1, the bit is valid; Reading this register gives the data at the top of the transmit FIFO or the data in the THR   |
| [7:0]     | RO | 8'b0  | Transmit FIFO Read               |  |   |
| 0x78      |    |       |                                  | RFW(Receive FIFO Write)                      | FIFO_ACCESS == YES  |
| [31:10]   | RO | 22'b0 | Reserved                         | Reserved                                     | FAR[0] = 1, the bit is valid; This bit is used to write framing error detection information to the receive FIFO or the RBR<br>FAR[0] = 1, the bit is valid; This bit is used to write parity error detection information to the receive FIFO or the RBR |
| [9]       | WO | 1'b0  | RFPE                             |  |   |
| [8]       | WO | 1'b0  | RFPE                             |  |   |

|        |    |                              |                     |   |
|--------|----|------------------------------|---------------------|---|
| [7:0]  | WO | 8'b0                         | RFWD                | FAR[0] = 1,the bit is valid;This bit of the data that is written to the RFWD is pushed into the receive FIFO or the RBR   |
| 0x7C   |    | USR(UART Status Register)    |                     |   |
| [31:5] | RO | 27'b0                        | Reserved            | Reserved  |
| [4]    | RO | 1'b0                         | RFF                 | FIFO_STAT == YES,the bit is vlid;This is used to indicate that the receive FIFO is completely full<br>1'b0: not full<br>1'b1: full  |
| [3]    | RO | 1'b0                         | RFNE                | FIFO_STAT == YES,the bit is vlid;This is used to indicate that the receive FIFO contains one or more entries<br>1'b0: empty<br>1'b1: not empty  |
| [2]    | RO | 1'b1                         | TFE                 | FIFO_STAT == YES,the bit is vlid;This is used to indicate that the transmit FIFO is completely empty<br>1'b0: not empty<br>1'b1: empty  |
| [1]    | RO | 1'b1                         | TFNF                | FIFO_STAT == YES,the bit is vlid;This is used to indicate that the transmit FIFO in not full<br>1'b0: full<br>1'b1: not full  |
| [0]    | RO | 1'b0                         | BUSY                | This is indicates that a serial transfer is in progress, when cleared indicates that the DW_apb_uart is idle or inactive<br>1'b0: idle or inactive<br>1'b1: busy (actively transferring data) |
| 0x80   |    | TFL(Transmit FIFO Level)     |                     | FIFO_STAT == YES;FIFO_ADDR_WIDTH=4  |
| [31:5] | RO | 27'b0                        | Reserved            | Reserved  |
| [4:0]  | RO | 5'b0                         | Transmit FIFO Level | This is indicates the number of data entries in the transmit FIFO   |
| 0x84   |    | RFL(Receive FIFO Level)      |                     | FIFO_STAT == YES;FIFO_ADDR_WIDTH=4  |
| [31:5] | RO | 27'b0                        | Reserved            | Reserved  |
| [4:0]  | RO | 5'b0                         | Receive FIFO Level  | This is indicates the number of data entries in the receive FIFO  |
| 0x88   |    | SRR(Software Reset Register) |                     | SHADOW == YES   |
| [31:3] | RO | 29'b0                        | Reserved            | Reserved  |
| [2]    | WO | 1'b0                         | XFR                 | FIFO_MODE == None,the written have no effect;XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit(FCR[2])   |
| [1]    | WO | 1'b0                         | RFR                 | FIFO_MODE == None,the written have no effect;RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit(FCR[1])   |

|        |    |                                     |                               |  |
|--------|----|-------------------------------------|-------------------------------|--|
| [0]    | WO | 1'b0                                | UR                            | This asynchronously resets the DW_apb_uart and synchronously removes the reset assertion   |
| 0x8C   |    | SRTS(Shadow Request to Send)        |                               | SHADOW == YES  |
| [31:1] | RO | 31'b0                               | Reserved                      | Reserved   |
| [0]    | RW | 1'b0                                | Shadow Request to Send        | This is a shadow register for the RTS bit(MCR[1])  |
| 0x90   |    | SBCR(Shadow Break Control Register) |                               | SHADOW == YES  |
| [31:1] | RO | 31'b0                               | Reserved                      | Reserved   |
| [0]    | RW | 1'b0                                | Shadow Break Control Register | This is a shadow register for the Break bit(LCR[6])  |
| 0x94   |    | SDMAM(Shadow DMA Mode)              |                               | FIFO_MODE != None and SHADOW == YES  |
| [31:1] | RO | 31'b0                               | Reserved                      | Reserved   |
| [0]    | RW | 1'b0                                | Shadow DMA Mode               | This is a shadow register for the DMA mode bit(FCR[3])<br>1'b0: mode 0<br>1'b1: mode 1   |
| 0x98   |    | SFE(Shadow FIFO Enable)             |                               | FIFO_MODE != None and SHADOW == YES  |
| [31:1] | RO | 31'b0                               | Reserved                      | Reserved   |
| [0]    | RW | 1'b0                                | Shadow FIFO Enable            | This is a shadow register for the FIFO enable bit(FCR[0])<br>1'b0: disable<br>1'b1: enable   |
| 0x9C   |    | SRT(Shadow RCVR Trigger)            |                               | FIFO_MODE != None and SHADOW == YES  |
| [31:2] | RO | 30'b0                               | Reserved                      | Reserved   |
| [1:0]  | RW | 2'b0                                | Shadow RCVR Trigger           | This is a shadow register for the RCVR trigger bits(FCR[7:6])<br>2'b00: 1 character in the FIFO<br>2'b01: FIFO ¼ full<br>2'b10: FIFO ½ full<br>2'b11: FIFO 2 less than full  |
| 0xA0   |    | STET(Shadow TX Empty Trigger)       |                               | FIFO_MODE != None and THRE_MODE_USER == Enabled and SHADOW == YES  |
| [31:2] | RO | 30'b0                               | Reserved                      | Reserved   |
| [1:0]  | RW | 2'b0                                | Shadow TX Empty Trigger       | THRE_MODE_USER == Disabled,the written have no effect;This is a shadow register for the TX empty trigger bits (FCR[5:4])<br>2'b00: FIFO empty<br>2'b01: 2 characters in the FIFO<br>2'b10: FIFO ¼ full<br>2'b11: FIFO ½ full |
| 0xA4   |    | HTX(Halt TX)                        |                               |  |
| [31:1] | RO | 31'b0                               | Reserved                      | Reserved   |

|         |    |                                   |                          |  |
|---------|----|-----------------------------------|--------------------------|--|
| [0]     | RW | 1'b0                              | Halt TX                  | FIFO_MODE == None,the written have no effect;This register is use to halt transmissions for testing<br>1'b0: disable<br>1'b1: enable                                 |
| 0xA8    |    | DMASA(DMA Software Acknowledge)   |                          |  |
| [31:1]  | RO | 31'b0                             | Reserved                 | Reserved   |
| [0]     | WO | 1'b0                              | DMA Software Acknowledge | DMA_EXTRA == No,the written have no effect;This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition |
| 0xF4    |    | CPR(Component Parameter Register) |                          |  |
| [31:24] | RO | 8'b0                              | Reserved                 | Reserved   |
| [23:16] | RO | 8'b0                              | FIFO_MODE                | 8'b0x00: 0<br>8'b0x01: 16<br>8'b0x02: 32<br>.....<br>8'b0x80: 2048<br>8'b0x81- 0xff: reserved  |
| [15:14] | RO | 2'b0                              | Reserved                 | Reserved   |
| [13]    | RO | 1'b0                              | DMA_EXTRA                | 1'b0: FALSE<br>1'b1: TRUE  |
| [12]    | RO | 1'b0                              | UART_ADD_ENCODED_PARAMS  | 1'b0: FALSE<br>1'b1: TRUE  |
| [11]    | RO | 1'b0                              | SHADOW                   | 1'b0: FALSE<br>1'b1: TRUE  |
| [10]    | RO | 1'b0                              | FIFO_STAT                | 1'b0: FALSE<br>1'b1: TRUE  |
| [9]     | RO | 1'b0                              | FIFO_ACCESS              | 1'b0: FALSE<br>1'b1: TRUE  |
| [8]     | RO | 1'b0                              | ADDITIONAL_FEAT          | 1'b0: FALSE<br>1'b1: TRUE  |
| [7]     | RO | 1'b0                              | SIR_LP_MODE              | 1'b0: FALSE<br>1'b1: TRUE  |
| [6]     | RO | 1'b0                              | SIR_MODE                 | 1'b0: FALSE<br>1'b1: TRUE  |
| [5]     | RO | 1'b0                              | THRE_MODE                | 1'b0: FALSE<br>1'b1: TRUE  |
| [4]     | RO | 1'b0                              | AFCE_MODE                | 1'b0: FALSE<br>1'b1: TRUE  |
| [3:2]   | RO | 2'b0                              | Reserved                 | Reserved   |
| [1:0]   | RO | 2'b0                              | APB_DATA_WIDTH           | 2'b00: 8 bits<br>2'b01: 16 bits<br>2'b10: 32 bits<br>2'b11: reserved   |
| 0xF8    |    | UCV(UART Component Version)       |                          |  |
|         |    | ADDITIONAL_FEATURES == YES        |                          |  |

|        |    |                |                              |   |
|--------|----|----------------|------------------------------|---|
| [31:0] | RO | 32'b0          | UART Component Version       | ASCII value for each number in the version, followed by *. For example 32_30_31_2A represents the version 2.01* |
| 0xFC   |    |                | CTR(Component Type Register) | ADDITIONAL_FEATURES == YES  |
| [31:0] | RO | 32'b0x44570110 | Peripheral ID                | This register contains the peripherals identification code  |

## 4.11 Pulse Width Modulation (PWM)

CMT4501 supports 6 channels of Pulse Width Modulation (PWM) outputs. PWM outputs generate waveforms with variable duty cycle or pulse width programmed by registers. And each of the 6 PWM outputs can be individually programmed. Their duty cycles are controlled by programming individual counters associated with each channel.

The master clock is 16MHz. For each PWM outputs, first there is a prescaler (pre-divider) with division ratio of 2 to 128 (only 2<sup>N</sup> division ratios are supported), followed by another 16bit counter with programmable max count, denoted as top\_count. When the 16bit counter counts from 0 to top\_count, it resets back to 0. So the frequency of the PWM is given by:

$$\text{Freq\_PWM} = 16\text{MHz} / (N\_prescaler * N\_top\_count);$$

A threshold counter number can be programmed, when the 16bit counter reaches the threshold, PWM output toggles. So the duty cycle is:

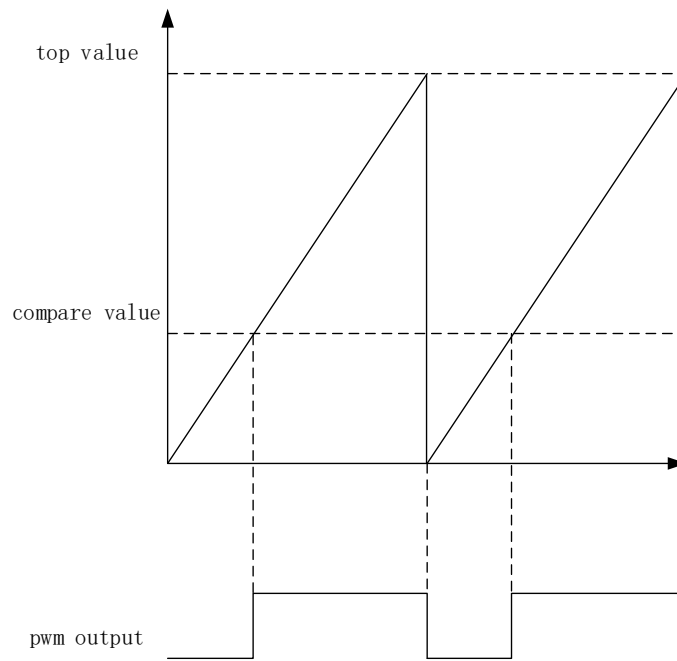
$$\text{Duty\_cycle\_PWM} = N\_threshold / N\_top\_count;$$

The polarity of the PWM can also be programmed, which indicates output 1 or 0 when counter is below/above the threshold. A PWM waveform vs counter values are illustrated in the following Figure 13, where the polarity is positive. Also in this case the counter ramps up and then resets, we call it “up mode”.

There is also a “up and down mode”, where the counter ramps up to count\_top and then ramps down, instead of reset.

As discussed above, the key register bits for one PWM channel are: 16bit top\_count, 16bit threshold count, 3bit prescaler count, PWM polarity, PWM mode (up or up/down), PWM enable, and PWM load enable (load new settings). All 6 PWM channels can be individually programmed by registers with addresses from 0x4000\_E004 to 0x4000\_E044. In addition, one should enable registers 0x4000\_E000<0><4> to allow all PWM channels can be programmed. For details please refer to documents of CMT4501 register tables.





**Figure13: PWM operation**

PWM related registers are listed below:

Base address: 4000\_E000

| OFFSET    | TYPE | RESET | NAME        | DESCRIPTION   |
|-----------|------|-------|-------------|---|
| 0x00      |      |       | PWMEN       | pwm enable  |
| [31 : 18] | RO   | 14'b0 | reserved    | Reserved  |
| [17]      | RW   | 1'b0  | pwm_load_45 | load parameter of PWM channel 4, 5. need to be conjunction with setting bit16 of PWMxCTL0 registers.<br>1'b0: no load<br>1'b1: load |
| [16]      | RW   | 1'b0  | pwm_en_45   | enable of PWM channel 4, 5. need to be conjunction with setting bit0 of PWMxCTL0 registers.<br>1'b0: disable<br>1'b1: enable        |
| [15]      | RW   | 1'b0  | pwm_load_23 | load parameter of PWM channel 2, 3. need to be conjunction with setting bit16 of PWMxCTL0 registers.<br>1'b0: no load<br>1'b1: load |
| [14]      | RW   | 1'b0  | pwm_en_23   | enable of PWM channel 2, 3. need to be conjunction with setting bit0 of PWMxCTL0 registers.<br>1'b0: disable<br>1'b1: enable        |
| [13]      | RW   | 1'b0  | pwm_load_01 | load parameter of PWM channel 0, 1. need to be conjunction with setting bit16 of PWMxCTL0 registers.<br>1'b0: no load<br>1'b1: load |

|          |    |      |              |   |
|----------|----|------|--------------|---|
| [12]     | RW | 1'b0 | pwm_en_01    | enable of PWM channel 0, 1. need to be conjunction with setting bit0 of PWMxCTL0 registers.<br>1'b0: disable<br>1'b1: enable            |
| [11]     | RW | 1'b0 | pwm_load_345 | load parameter of PWM channel 3, 4, 5. need to be conjunction with setting bit16 of PWMxCTL0 registers.<br>1'b0: no load<br>1'b1: load  |
| [10]     | RW | 1'b0 | pwm_en_345   | enable of PWM channel 3, 4, 5. need to be conjunction with setting bit0 of PWMxCTL0 registers.<br>1'b0: disable<br>1'b1: enable         |
| [9]      | RW | 1'b0 | pwm_load_012 | load parameter of PWM channel 0, 1, 2. need to be conjunction with setting bit16 of PWMxCTL0 registers.<br>1'b0: no load<br>1'b1: load  |
| [8]      | RW | 1'b0 | pwm_en_012   | enable of PWM channel 0, 1, 2. need to be conjunction with setting bit0 of PWMxCTL0 registers.<br>1'b0: disable<br>1'b1: enable         |
| [ 7 : 5] | RO | 3'b0 | reserved     | Reserved  |
| [4]      | RW | 1'b0 | pwm_load_all | load parameter of all six PWM channels. need to be conjunction with setting bit16 of PWMxCTL0 registers.<br>1'b0: no load<br>1'b1: load |
| [ 3 : 1] | RO | 3'b0 | reserved     | Reserved  |
| [0]      | RW | 1'b0 | pwm_en_all   | enable of all six PWM channels. need to be conjunction with setting bit0 of PWMxCTL0 registers.<br>1'b0: disable<br>1'b1: enable        |

|      |          |                                |
|------|----------|--------------------------------|
| 0x04 | PWM0CTL0 | pwm channel 0 contrl reigister |
|------|----------|--------------------------------|

|           |    |       |                   |   |
|-----------|----|-------|-------------------|---|
| [31]      | RW | 1'b0  | pwm0_load_instant | instant load parameter of PWM channel 0.<br>1'b0: no load<br>1'b1: instant load |
| [30 : 17] | RO | 14'b0 | reserved          | Reserved  |
| [16]      | RW | 1'b0  | pwm0_load         | load parameter of PWM channel 0.<br>1'b0: no load<br>1'b1: load                 |
| [15]      | RO | 1'b0  | reserved          | Reserved  |

|           |    |      |               |  |
|-----------|----|------|---------------|--|
| [14 : 12] | RW | 3'b0 | pwm0_clk_div  | clock prescaler of PWM channel 0.<br>3'b000: pwm_clk is divided by 1 for count clock<br>3'b001: pwm_clk is divided by 2 for count clock<br>3'b010: pwm_clk is divided by 4 for count clock<br>3'b011: pwm_clk is divided by 8 for count clock<br>3'b100: pwm_clk is divided by 16 for count clock<br>3'b101: pwm_clk is divided by 32 for count clock<br>3'b110: pwm_clk is divided by 64 for count clock<br>3'b111: pwm_clk is divided by 128 for count clock |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
| [11 : 9]  | RO | 3'b0 | reserved      | Reserved   |
| [8]       | RW | 1'b0 | pwm0_cnt_mode | count mode of PWM channel 0.<br>1'b0: up mode<br>1'b1: up and down mode  |
| [ 7 : 5]  | RO | 3'b0 | reserved      | Reserved   |
| [4]       | RW | 1'b0 | pwm0_polarity | output polarity setting of PWM channel 0.<br>1'b0: rising edge. Second edge within the PWM period is rising<br>1'b1: falling edge. Second edge within the PWM period is falling  |
| [ 3 : 1]  | RO | 3'b0 | reserved      | Reserved   |
| [0]       | RW | 1'b0 | pwm0_en       | enable of PWM channel 0.<br>1'b0: disable<br>1'b1: enable  |

| 0x08      |    |       | PWM0CTL1     | pwm channel 0 conter value setting     |
|-----------|----|-------|--------------|--|
| [31 : 16] | RW | 16'b0 | pwm0_cmp_val | the compare value of PWM channel 0     |
| [15 : 0]  | RW | 16'b0 | pwm0_cnt_top | the counter top value of PWM channel 0 |

| 0x10      |    |       | PWM1CTL0          | pwm channel 1 contrl reigister  |
|-----------|----|-------|-------------------|---|
| [31]      | RW | 1'b0  | pwm1_load_instant | instant load parameter of PWM channel 1.<br>1'b0: no load<br>1'b1: instant load |
| [30 : 17] | RO | 14'b0 | reserved          | Reserved  |
| [16]      | RW | 1'b0  | pwm1_load         | load parameter of PWM channel 1.<br>1'b0: no load<br>1'b1: load                 |
| [15]      | RO | 1'b0  | reserved          | Reserved  |

|           |    |      |               |  |
|-----------|----|------|---------------|--|
| [14 : 12] | RW | 3'b0 | pwm1_clk_div  | clock prescaler of PWM channel 1.<br>3'b000: pwm_clk is divided by 1 for count clock<br>3'b001: pwm_clk is divided by 2 for count clock<br>3'b010: pwm_clk is divided by 4 for count clock<br>3'b011: pwm_clk is divided by 8 for count clock<br>3'b100: pwm_clk is divided by 16 for count clock<br>3'b101: pwm_clk is divided by 32 for count clock<br>3'b110: pwm_clk is divided by 64 for count clock<br>3'b111: pwm_clk is divided by 128 for count clock |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
| [11 : 9]  | RO | 3'b0 | reserved      | Reserved   |
| [8]       | RW | 1'b0 | pwm1_cnt_mode | count mode of PWM channel 1.<br>1'b0: up mode<br>1'b1: up and down mode  |
| [ 7 : 5]  | RO | 3'b0 | reserved      | Reserved   |
| [4]       | RW | 1'b0 | pwm1_polarity | output polarity setting of PWM channel 1.<br>1'b0: rising edge. Second edge within the PWM period is rising<br>1'b1: falling edge. Second edge within the PWM period is falling  |
| [ 3 : 1]  | RO | 3'b0 | reserved      | Reserved   |
| [0]       | RW | 1'b0 | pwm1_en       | enable of PWM channel 1.<br>1'b0: disable<br>1'b1: enable  |

| 0x14 | PWM1CTL1 | pwm channel 1 conter value setting |
|------|----------|------------------------------------|
|------|----------|------------------------------------|

|           |    |       |              |  |
|-----------|----|-------|--------------|--|
| [31 : 16] | RW | 16'b0 | pwm1_cmp_val | the compare value of PWM channel 1     |
| [15 : 0]  | RW | 16'b0 | pwm1_cnt_top | the counter top value of PWM channel 1 |

| 0x1C | PWM2CTL0 | pwm channel 2 contrl reigister |
|------|----------|--------------------------------|
|------|----------|--------------------------------|

|           |    |       |                   |   |
|-----------|----|-------|-------------------|---|
| [31]      | RW | 1'b0  | pwm2_load_instant | instant load parameter of PWM channel 2.<br>1'b0: no load<br>1'b1: instant load |
| [30 : 17] | RO | 14'b0 | reserved          | Reserved  |
| [16]      | RW | 1'b0  | pwm2_load         | load parameter of PWM channel 2.<br>1'b0: no load<br>1'b1: load                 |
| [15]      | RO | 1'b0  | reserved          | Reserved  |

|           |    |      |               |  |
|-----------|----|------|---------------|--|
| [14 : 12] | RW | 3'b0 | pwm2_clk_div  | clock prescaler of PWM channel 2.<br>3'b000: pwm_clk is divided by 1 for count clock<br>3'b001: pwm_clk is divided by 2 for count clock<br>3'b010: pwm_clk is divided by 4 for count clock<br>3'b011: pwm_clk is divided by 8 for count clock<br>3'b100: pwm_clk is divided by 16 for count clock<br>3'b101: pwm_clk is divided by 32 for count clock<br>3'b110: pwm_clk is divided by 64 for count clock<br>3'b111: pwm_clk is divided by 128 for count clock |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
| [11 : 9]  | RO | 3'b0 | reserved      | Reserved   |
| [8]       | RW | 1'b0 | pwm2_cnt_mode | count mode of PWM channel 2.<br>1'b0: up mode<br>1'b1: up and down mode  |
| [ 7 : 5]  | RO | 3'b0 | reserved      | Reserved   |
| [4]       | RW | 1'b0 | pwm2_polarity | output polarity setting of PWM channel 2.<br>1'b0: rising edge. Second edge within the PWM period is rising<br>1'b1: falling edge. Second edge within the PWM period is falling  |
| [ 3 : 1]  | RO | 3'b0 | reserved      | Reserved   |
| [0]       | RW | 1'b0 | pwm2_en       | enable of PWM channel 2.<br>1'b0: disable<br>1'b1: enable  |

| 0x20 | PWM2CTL1 | pwm channel 2 conter value setting |
|------|----------|------------------------------------|
|------|----------|------------------------------------|

|           |    |       |              |  |
|-----------|----|-------|--------------|--|
| [31 : 16] | RW | 16'b0 | pwm2_cmp_val | the compare value of PWM channel 2     |
| [15 : 0]  | RW | 16'b0 | pwm2_cnt_top | the counter top value of PWM channel 2 |

| 0x28 | PWM3CTL0 | pwm channel 3 contrl reigister |
|------|----------|--------------------------------|
|------|----------|--------------------------------|

|           |    |       |                   |   |
|-----------|----|-------|-------------------|---|
| [31]      | RW | 1'b0  | pwm3_load_instant | instant load parameter of PWM channel 3.<br>1'b0: no load<br>1'b1: instant load |
| [30 : 17] | RO | 14'b0 | reserved          | Reserved  |
| [16]      | RW | 1'b0  | pwm3_load         | load parameter of PWM channel 3.<br>1'b0: no load<br>1'b1: load                 |
| [15]      | RO | 1'b0  | reserved          | Reserved  |

|           |    |      |               |  |
|-----------|----|------|---------------|--|
| [14 : 12] | RW | 3'b0 | pwm3_clk_div  | clock prescaler of PWM channel 3.<br>3'b000: pwm_clk is divided by 1 for count clock<br>3'b001: pwm_clk is divided by 2 for count clock<br>3'b010: pwm_clk is divided by 4 for count clock<br>3'b011: pwm_clk is divided by 8 for count clock<br>3'b100: pwm_clk is divided by 16 for count clock<br>3'b101: pwm_clk is divided by 32 for count clock<br>3'b110: pwm_clk is divided by 64 for count clock<br>3'b111: pwm_clk is divided by 128 for count clock |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
| [11 : 9]  | RO | 3'b0 | reserved      | Reserved   |
| [8]       | RW | 1'b0 | pwm3_cnt_mode | count mode of PWM channel 3.<br>1'b0: up mode<br>1'b1: up and down mode  |
| [ 7 : 5]  | RO | 3'b0 | reserved      | Reserved   |
| [4]       | RW | 1'b0 | pwm3_polarity | output polarity setting of PWM channel 3.<br>1'b0: rising edge. Second edge within the PWM period is rising<br>1'b1: falling edge. Second edge within the PWM period is falling  |
| [ 3 : 1]  | RO | 3'b0 | reserved      | Reserved   |
| [0]       | RW | 1'b0 | pwm3_en       | enable of PWM channel 3.<br>1'b0: disable<br>1'b1: enable  |

| 0x2C      |    |       | PWM3CTL1     | pwm channel 0 conter value setting     |
|-----------|----|-------|--------------|--|
| [31 : 16] | RW | 16'b0 | pwm3_cmp_val | the compare value of PWM channel 3     |
| [15 : 0]  | RW | 16'b0 | pwm3_cnt_top | the counter top value of PWM channel 3 |

| 0x34      |    |       | PWM4CTL0          | pwm channel 4 contrl reigister  |
|-----------|----|-------|-------------------|---|
| [31]      | RW | 1'b0  | pwm4_load_instant | instant load parameter of PWM channel 4.<br>1'b0: no load<br>1'b1: instant load |
| [30 : 17] | RO | 14'b0 | reserved          | Reserved  |
| [16]      | RW | 1'b0  | pwm4_load         | load parameter of PWM channel 4.<br>1'b0: no load<br>1'b1: load                 |
| [15]      | RO | 1'b0  | reserved          | Reserved  |

|           |    |      |               |  |
|-----------|----|------|---------------|--|
| [14 : 12] | RW | 3'b0 | pwm4_clk_div  | clock prescaler of PWM channel 4.<br>3'b000: pwm_clk is divided by 1 for count clock<br>3'b001: pwm_clk is divided by 2 for count clock<br>3'b010: pwm_clk is divided by 4 for count clock<br>3'b011: pwm_clk is divided by 8 for count clock<br>3'b100: pwm_clk is divided by 16 for count clock<br>3'b101: pwm_clk is divided by 32 for count clock<br>3'b110: pwm_clk is divided by 64 for count clock<br>3'b111: pwm_clk is divided by 128 for count clock |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
| [11 : 9]  | RO | 3'b0 | reserved      | Reserved   |
| [8]       | RW | 1'b0 | pwm4_cnt_mode | count mode of PWM channel 4.<br>1'b0: up mode<br>1'b1: up and down mode  |
| [ 7 : 5]  | RO | 3'b0 | reserved      | Reserved   |
| [4]       | RW | 1'b0 | pwm4_polarity | output polarity setting of PWM channel 4.<br>1'b0: rising edge. Second edge within the PWM period is rising<br>1'b1: falling edge. Second edge within the PWM period is falling  |
| [ 3 : 1]  | RO | 3'b0 | reserved      | Reserved   |
| [0]       | RW | 1'b0 | pwm4_en       | enable of PWM channel 4.<br>1'b0: disable<br>1'b1: enable  |

| 0x38 | PWM4CTL1 | pwm channel 4 conter value setting |
|------|----------|------------------------------------|
|------|----------|------------------------------------|

|           |    |       |              |  |
|-----------|----|-------|--------------|--|
| [31 : 16] | RW | 16'b0 | pwm4_cmp_val | the compare value of PWM channel 4     |
| [15 : 0]  | RW | 16'b0 | pwm4_cnt_top | the counter top value of PWM channel 4 |

| 0x40 | PWM5CTL0 | pwm channel 5 contrl reigister |
|------|----------|--------------------------------|
|------|----------|--------------------------------|

|           |    |       |                   |   |
|-----------|----|-------|-------------------|---|
| [31]      | RW | 1'b0  | pwm5_load_instant | instant load parameter of PWM channel 5.<br>1'b0: no load<br>1'b1: instant load |
| [30 : 17] | RO | 14'b0 | reserved          | Reserved  |
| [16]      | RW | 1'b0  | pwm5_load         | load parameter of PWM channel 5.<br>1'b0: no load<br>1'b1: load                 |
| [15]      | RO | 1'b0  | reserved          | Reserved  |

|           |    |      |               |  |
|-----------|----|------|---------------|--|
| [14 : 12] | RW | 3'b0 | pwm5_clk_div  | clock prescaler of PWM channel 5.<br>3'b000: pwm_clk is divided by 1 for count clock<br>3'b001: pwm_clk is divided by 2 for count clock<br>3'b010: pwm_clk is divided by 4 for count clock<br>3'b011: pwm_clk is divided by 8 for count clock<br>3'b100: pwm_clk is divided by 16 for count clock<br>3'b101: pwm_clk is divided by 32 for count clock<br>3'b110: pwm_clk is divided by 64 for count clock<br>3'b111: pwm_clk is divided by 128 for count clock |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
|           |    |      |               |  |
| [11 : 9]  | RO | 3'b0 | reserved      | Reserved   |
| [8]       | RW | 1'b0 | pwm5_cnt_mode | count mode of PWM channel 5.<br>1'b0: up mode<br>1'b1: up and down mode  |
| [ 7 : 5]  | RO | 3'b0 | reserved      | Reserved   |
| [4]       | RW | 1'b0 | pwm5_polarity | output polarity setting of PWM channel 5.<br>1'b0: rising edge. Second edge within the PWM period is rising<br>1'b1: falling edge. Second edge within the PWM period is falling  |
| [ 3 : 1]  | RO | 3'b0 | reserved      | Reserved   |
| [0]       | RW | 1'b0 | pwm5_en       | enable of PWM channel 5.<br>1'b0: disable<br>1'b1: enable  |

|           |          |       |                                    |  |
|-----------|----------|-------|------------------------------------|--|
| 0x44      | PWM5CTL1 |       | pwm channel 5 conter value setting |  |
| [31 : 16] | RW       | 16'b0 | pwm5_cmp_val                       | the compare value of PWM channel 5     |
| [15 : 0]  | RW       | 16'b0 | pwm5_cnt_top                       | the counter top value of PWM channel 5 |

## 4.12 Quadrature Decoder (QDEC)

The quadrature decoder provides buffered decoding of quadrature-encoded sensor signals with input debounce filters. It is suitable for mechanical and optical sensors. The sample period and accumulation are configurable to match application requirements. The quadrature decoder has three-axis capability and index channel support. It can be programmed as 4x/2x/1x count mode.

Quadrature decoder related registers are listed below:

Base address: 4000\_B000

| OFFSET | TYPE       | RESET | NAME     | DESCRIPTION      |
|--------|------------|-------|----------|------------------|
| 0x00   |            |       |          |                  |
| [31:9] | —          | 23'b0 | reserved |                  |
| [8]    | RW         | 1'b0  | chnz_en  | enable channel z |
| [7:5]  | —          | 3'b0  | reserved |                  |
| [4]    | RW         | 1'b0  | chny_en  | enable channel y |
| [3:1]  | —          | 3'b0  | reserved |                  |
| [0]    | RW         | 1'b0  | chnx_en  | enable channel x |
| 0x04   | int_enable |       |          |                  |



|         |    |      |                 |  |
|---------|----|------|-----------------|--|
| [31:30] | —  | 2'b0 | reserved        |  |
| [29]    | RW | 1'b0 | int_quaz_02f_en | enable interrupt, counter addition overflow<br>( from 0 to F)            |
| [28]    | RW | 1'b0 | int_quaz_f20_en | enable interrupt, counter subtraction overflow<br>( from F to 0)         |
| [27]    | RW | 1'b0 | int_quay_02f_en |  |
| [26]    | RW | 1'b0 | int_quay_f20_en |  |
| [25]    | RW | 1'b0 | int_quax_02f_en |  |
| [24]    | RW | 1'b0 | int_quax_f20_en |  |
| [23]    | —  | 1'b0 | reserved        |  |
| [22]    | RW | 1'b0 | incz_int_mode   | index counter interrupt mode<br>0 index changes, 1 index equals hit      |
| [21]    | —  | 1'b0 | reserved        |  |
| [20]    | RW | 1'b0 | int_incz_en     | enable index counter interrupt   |
| [19]    | —  | 1'b0 | reserved        |  |
| [18]    | RW | 1'b0 | quaz_int_mode   | quadrature counter interrupt mode<br>0 index changes, 1 index equals hit |
| [17]    | —  | 1'b0 | reserved        |  |
| [16]    | RW | 1'b0 | int_quaz_en     | enable quadrature counter interrupt                                      |
| [15]    | —  | 1'b0 | reserved        |  |
| [14]    | RW | 1'b0 | incy_int_mode   |  |
| [13]    | —  | 1'b0 | reserved        |  |
| [12]    | RW | 1'b0 | int_incy_en     |  |
| [11]    | —  | 1'b0 | reserved        |  |
| [10]    | RW | 1'b0 | quay_int_mode   |  |
| [9]     | —  | 1'b0 | reserved        |  |
| [8]     | RW | 1'b0 | int_quay_en     |  |
| [7]     | —  | 1'b0 | reserved        |  |
| [6]     | RW | 1'b0 | incx_int_mode   |  |
| [5]     | —  | 1'b0 | reserved        |  |
| [4]     | RW | 1'b0 | int_incx_mode   |  |
| [3]     | —  | 1'b0 | reserved        |  |
| [2]     | RW | 1'b0 | quax_int_mode   |  |
| [1]     | —  | 1'b0 | reserved        |  |
| [0]     | RW | 1'b0 | int_quax_en     |  |

0x08

int\_clear

|         |    |      |              |                               |
|---------|----|------|--------------|-------------------------------|
| [31:30] | —  | 2'b0 | reserved     |                               |
| [29]    | WC | 1'b0 | quaz_02f_clr | clear 0 to F interrupt        |
| [28]    | WC | 1'b0 | quaz_f20_clr | clear F to 0 interrupt        |
| [27]    | WC | 1'b0 | quay_02f_clr |                               |
| [26]    | WC | 1'b0 | quay_f20_clr |                               |
| [25]    | WC | 1'b0 | quax_02f_clr |                               |
| [24]    | WC | 1'b0 | quax_f20_clr |                               |
| [23:21] | —  | 3'b0 | reserved     |                               |
| [20]    | WC | 1'b0 | incz_clr     | clear index counter interrupt |

|         |    |      |          |                                    |
|---------|----|------|----------|------------------------------------|
| [19:17] | —  | 3'b0 | reserved |                                    |
| [16]    | WC | 1'b0 | quaz_clr | clear quadrature counter interrupt |
| [15:13] | —  | 3'b0 | reserved |                                    |
| [12]    | WC | 1'b0 | incy_clr |                                    |
| [11:9]  | —  | 3'b0 | reserved |                                    |
| [8]     | WC | 1'b0 | quay_clr |                                    |
| [7:5]   | —  | 3'b0 | reserved |                                    |
| [4]     | WC | 1'b0 | incx_clr |                                    |
| [3:1]   | —  | 3'b0 | reserved |                                    |
| [0]     | WC | 1'b0 | quax_clr |                                    |

| 0x0C    |    | int_status |              |                                     |
|---------|----|------------|--------------|-------------------------------------|
| [31:30] | —  | 2'b0       | reserved     |                                     |
| [29]    | RO | 1'b0       | int_quaz_02f | 0 to F interrupt status             |
| [28]    | RO | 1'b0       | int_quaz_f20 | F to 0 interrupt status             |
| [27]    | RO | 1'b0       | int_quay_02f |                                     |
| [26]    | RO | 1'b0       | int_quay_f20 |                                     |
| [25]    | RO | 1'b0       | int_quax_02f |                                     |
| [24]    | RO | 1'b0       | int_quax_f20 |                                     |
| [23:21] | —  | 3'b0       | reserved     |                                     |
| [20]    | RO | 1'b0       | int_inc_z    | index counter interrupt status      |
| [19:17] | —  | 3'b0       | reserved     |                                     |
| [16]    | RO | 1'b0       | int_qua_z    | quadrature counter interrupt status |
| [15:13] | —  | 3'b0       | reserved     |                                     |
| [12]    | RO | 1'b0       | int_inc_y    |                                     |
| [11:9]  | —  | 3'b0       | reserved     |                                     |
| [8]     | RO | 1'b0       | int_qua_y    |                                     |
| [7:5]   | —  | 3'b0       | reserved     |                                     |
| [4]     | RO | 1'b0       | int_inc_x    |                                     |
| [3:1]   | —  | 3'b0       | reserved     |                                     |
| [0]     | RO | 1'b0       | int_qua_x    |                                     |

| 0x10    |    |       |           |  |
|---------|----|-------|-----------|--|
| [31:18] | —  | 14'b0 | reserved  |  |
| [17:16] | RW | 2'b0  | incx_mode | index counter mode<br>00 high level 01 positive edge<br>10 negative edge 11 pos and neg edge |
| [15:2]  | —  | 14'b0 | reserved  |  |
| [1:0]   | RW | 2'b0  | quax_mode | quadrature counter mode<br>01 mode 1x , 10 mode 2x, 11 mode 3x                               |

| 0x14   |    |       |          |  |
|--------|----|-------|----------|--|
| [31:0] | RW | 32'b0 | quax_hit | to compare with qua_cnt, trigger interrupt |

| 0x18   |    |       |          |  |
|--------|----|-------|----------|--|
| [31:0] | RW | 32'b0 | incx_hit | to compare with inc_cnt, trigger interrupt |

**0x1C**

|        |    |       |          |                    |
|--------|----|-------|----------|--------------------|
| [31:0] | RO | 32'b0 | quax_cnt | quadrature counter |
|--------|----|-------|----------|--------------------|

**0x20**

|        |    |       |          |               |
|--------|----|-------|----------|---------------|
| [31:0] | RO | 32'b0 | incx_cnt | index counter |
|--------|----|-------|----------|---------------|

**0x24**

|         |    |       |           |
|---------|----|-------|-----------|
| [31:18] | —  | 14'b0 | reserved  |
| [17:16] | RW | 2'b0  | incy_mode |
| [15:2]  | —  | 14'b0 | reserved  |
| [1:0]   | RW | 2'b0  | quay_mode |

**0x28**

|        |    |       |          |
|--------|----|-------|----------|
| [31:0] | RW | 32'b0 | quay_hit |
|--------|----|-------|----------|

**0x2C**

|        |    |       |          |
|--------|----|-------|----------|
| [31:0] | RW | 32'b0 | incy_hit |
|--------|----|-------|----------|

**0x30**

|        |    |       |          |
|--------|----|-------|----------|
| [31:0] | RO | 32'b0 | quay_cnt |
|--------|----|-------|----------|

**0x34**

|        |    |       |          |
|--------|----|-------|----------|
| [31:0] | RO | 32'b0 | incy_cnt |
|--------|----|-------|----------|

**0x38**

|         |    |       |           |
|---------|----|-------|-----------|
| [31:18] | —  | 14'b0 | reserved  |
| [17:16] | RW | 2'b0  | incz_mode |
| [15:2]  | —  | 14'b0 | reserved  |
| [1:0]   | RW | 2'b0  | quaz_mode |

**0x3C**

|        |    |       |          |
|--------|----|-------|----------|
| [31:0] | RW | 32'b0 | quaz_hit |
|--------|----|-------|----------|

**0x40**

|        |    |       |          |
|--------|----|-------|----------|
| [31:0] | RW | 32'b0 | incz_hit |
|--------|----|-------|----------|

**0x44**

|        |    |       |          |
|--------|----|-------|----------|
| [31:0] | RO | 32'b0 | quaz_cnt |
|--------|----|-------|----------|

**0x48**

|        |    |       |          |
|--------|----|-------|----------|
| [31:0] | RO | 32'b0 | incz_cnt |
|--------|----|-------|----------|

**0x3FC**

|        |    |       |       |
|--------|----|-------|-------|
| [31:0] | RW | 32'b0 | dummy |
|--------|----|-------|-------|

## 4.13 Key Scan (KSCAN)

Keyscan supports key matrix with upto 16 rows by 18 columns. Each individual rows or columns can be enabled or disabled through register settings. GPIO pins can be configured to be used for key scan. A few key scan Parameters can be set through registers, including polarity (low or high indicating key pressed); support multi-key-press or only single-key-press; de-bounce time (the time duration a key press is deemed valid) from 0 to 128mS with 255us step.

A valid key press can trigger an interrupt when keyscan interrupt is enabled. After a keyscan interrupt is serviced, writing 1 to the interrupt state register bit can clear the state bit.

The keyscan has a manual mode and an auto mode. For manual mode, when a keyscan interrupt is received, it is up to the MCU/software to scan the keyscan output pins and check the input pins, to determine which keys have been pressed. Manual mode is relatively slow and need CPU to process. On the contrary, in automode keyscan will automatically scan the output/input pins, and store the row/column info corresponding to the key pressed into read only registers, then trigger an interrupt for software to retrieve key press information.

Key scan related registers are listed below.

Base address: 4002\_4000

| OFFSET  | TYPE | RESET    | NAME     | DESCRIPTION  |
|---------|------|----------|----------|--|
| 0xC0    |      |          |          |  |
| [31:24] | RW   | 8'h00    | mkdi     | key scan debounce interval, 0-255, unit: 512uS   |
| [23]    | RW   | 1'b0     | mk_pol   | key matrix polarity, 0: active scan high, active sense high; 1: active scan low, active sense low; |
| [22]    | RO   | 1'b0     | reserved | no use/as  |
| [21]    | RW   | 1'b0     | asact    | auto scan on activity: 0, no auto scan, 1, auto scan on activity                                   |
| [20]    | RW   | 1'b0     | imkp     | ignore multi key press   |
| [19:2]  | RW   | 18'h0    | ms       | matrix scan outputs enable: 1: enable, 0: disable  |
| [1]     | RW   | 1'b0     | ks_ie    | key scan interrupt enable  |
| [0]     | RW   | 1'b0     | ks_en    | key scan enable  |
| 0xC4    |      |          |          |  |
| [31:18] | RO   | 14'b0    | reserved |  |
| [17]    | WC   | 1'b0     | mkp      | key pressed indicator, 0: no key press, 1: key pressed, write 1 to clear                           |
| [16:1]  | RO   | 16'h0FFF | mr       | key scan inputs states   |
| [0]     | WC   | 1'b0     | mi       | interrupt state, write 1 to clear interrupt, 0: no interrupt, 1: interrupt issued,                 |
| 0xC8    |      |          |          |  |
| [31:13] | RO   | 19'b0    | reserved |  |
| [12]    | RO   | 1'b0     | so       | scan on: 1: auto scan is on going, 0: scan off   |
| [11:10] | RO   | 2'b0     | mukp     | multi key pressed, 00, no key press, 01: 1 key press, 10, more than 1 key pressed                  |
| [9:5]   | RO   | 5'h1F    | rp       | row of key pressed, only for 1 key pressed case  |
| [4:0]   | RO   | 5'h1F    | cp       | column of key pressed, only for 1 key pressed case   |
| 0xCC    |      |          |          |  |
| [31:16] | RO   | 16'h0    | mkc1     | column 1 key pressed, for multi key pressed case   |
| [15:0]  | RO   | 16'h0    | mkc0     | column 0 key pressed, for multi key pressed case   |

|             |    |          |           |   |
|-------------|----|----------|-----------|---|
| <b>0xD0</b> |    |          |           |   |
| [31:16]     | RO | 16'h0    | mkc3      | column 3 key pressed, for multi key pressed case      |
| [15:0]      | RO | 16'h0    | mkc2      | column 2 key pressed, for multi key pressed case      |
| <b>0xD4</b> |    |          |           |   |
| [31:16]     | RO | 16'h0    | mkc5      | column 5 key pressed, for multi key pressed case      |
| [15:0]      | RO | 16'h0    | mkc4      | column 4 key pressed, for multi key pressed case      |
| <b>0xD8</b> |    |          |           |   |
| [31:16]     | RO | 16'h0    | mkc7      | column 7 key pressed, for multi key pressed case      |
| [15:0]      | RO | 16'h0    | mkc6      | column 6 key pressed, for multi key pressed case      |
| <b>0xDC</b> |    |          |           |   |
| [31:16]     | RO | 16'h0    | mkc9      | column 9 key pressed, for multi key pressed case      |
| [15:0]      | RO | 16'h0    | mkc8      | column 8 key pressed, for multi key pressed case      |
| <b>0xE0</b> |    |          |           |   |
| [31:16]     | RO | 16'h0    | mkc11     | column 11 key pressed, for multi key pressed case     |
| [15:0]      | RO | 16'h0    | mkc10     | column 10 key pressed, for multi key pressed case     |
| <b>0xE4</b> |    |          |           |   |
| [31:16]     | RO | 16'h0    | mkc13     | column 13 key pressed, for multi key pressed case     |
| [15:0]      | RO | 16'h0    | mkc12     | column 12 key pressed, for multi key pressed case     |
| <b>0xE8</b> |    |          |           |   |
| [31:16]     | RO | 16'h0    | mkc15     | column 15 key pressed, for multi key pressed case     |
| [15:0]      | RO | 16'h0    | mkc14     | column 14 key pressed, for multi key pressed case     |
| <b>0xEC</b> |    |          |           |   |
| [31:16]     | RO | 16'h0    | mkc17     | column 17 key pressed, for multi key pressed case     |
| [15:0]      | RO | 16'h0    | mkc16     | column 16 key pressed, for multi key pressed case     |
| <b>0xF0</b> |    |          |           |   |
| [31:16]     | RW | 16'h0    | reserved  |   |
| [15:0]      | RW | 16'h0FFF | mk_in_en  | enable/disable key scan inputs: 0: disable, 1: enable |
| <b>0xF4</b> |    |          |           |   |
| [31:2]      | RW | 30'h0    | reserved  |   |
| [1:0]       | RW | 2'b0     | ks_pena_i |   |
| <b>0xF8</b> |    |          |           |   |
| [31:0]      | RW | 32'h0    | ks_iosel  |   |

## 4.14 Analog to Digital Converter (ADC) with Programmable Gain Amplifier (PGA)

The 12bit SAR ADC has total 10 inputs. Among them, there are two for PGA inputs, and two differential inputs for the on-chip temperature sensor. The other six inputs can be programmed to 3 pair differential inputs or six single-ended inputs. There is a manual mode with which the ADC can be configured to convert a specific input in single-ended or differential and with a specific ADC

clock rate. There is also an auto sweep mode, namely all enabled input channels can be swept automatically in order by the ADC and the converted data will be stored at corresponding memory locations.

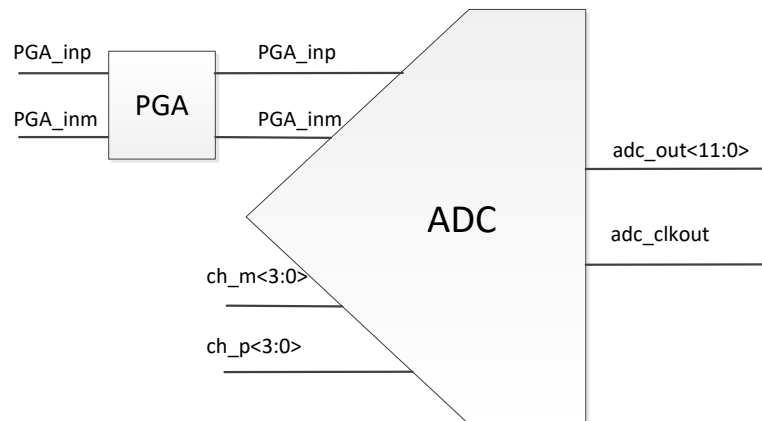


Figure 12: ADC

#### 4.14.1 PGA Path

The PGA provides 42dB gain range from 0dB to 42dB in 3dB steps.

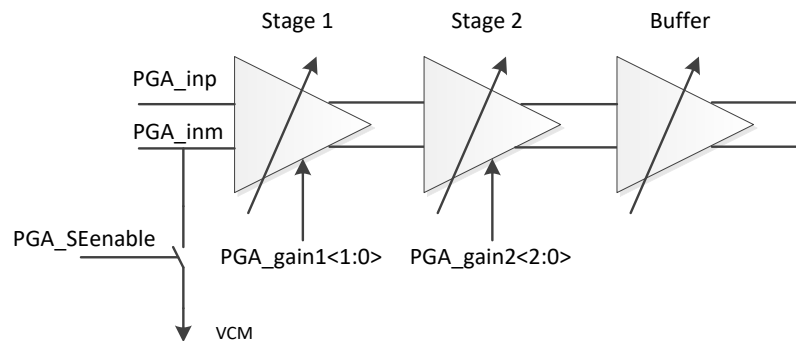


Figure 13: PGA path

| pga_gain1<1> | pga_gain1<0> | Stage1 gain (dB) | pga_gain2<2> | pga_gain2<1> | pga_gain2<0> | Stage2 gain (dB) |
|--------------|--------------|------------------|--------------|--------------|--------------|------------------|
| 0            | 0            | 0                | 0            | 0            | 0            | 0                |
| 0            | 1            | 12               | 0            | 0            | 1            | 3                |
| 1            | 0            | 24               | 0            | 1            | 0            | 6                |
|              |              |                  | 0            | 1            | 1            | 9                |
|              |              |                  | 1            | 0            | 0            | 12               |
|              |              |                  | 1            | 0            | 1            | 15               |
|              |              |                  | 1            | 1            | 0            | 18               |

Table 12: PGA gain

Set PGA\_SEenable to "1", PGA will be set to Single-ended mode by pulling the PGA into its Common-mode voltage.

#### 4.14.2 ADC Path

By default the ADC is configured in manual mode. In this mode, the ADC clock rate can be configured to 80k/160k/320k sample per second. Select the pair of inputs and configure it to differential or singled-ended (positive or negative). By default it is differential. After enabling, the ADC will take samples with the configured clock rate and store the data to a channel dependent memory location. For each channel a memory size of 128Byte is allocated, when it is full an interrupt bit will be flagged. Each sample of 12bits takes 2 Byte memory space.

| 0x4000_F07C |                    | Register Description  |
|-------------|--------------------|---|
| [4]         | adc_ctrl_override  | Set manual mode: 1: manual, 0: auto. Default 1                    |
| [3]         | adc_tconv_sel      | For auto mode only, adc conversion time sel: 0: 1.56us, 1: 2.34us |
| [2:1]       | adc_clk_sel        | For manual mode only, clkssel: 00: 80k, 01: 160k, 10: 320k        |
| [0]         | max_rate_256k_320k | For auto mode only, max rate base: 0, 256k, 1, 320k               |

| 0x4000_F048 |                   | Register Description  |
|-------------|-------------------|---|
| [11]        | adc12b_semode_enm | For manual mode only: 12 bit ADC single-ended mode negative side enable. Bit<11> Bit<8> cannot both be 1;<br>1: Enable single-ended mode<br>0: Differential mode  |
| [8]         | Adc12b_semode_epm | For manual mode only: 12 bit ADC single-ended mode positive side enable. Bit<8> Bit<11> cannot both be 1;<br>1: Enable single-ended mode<br>0: Differential mode  |
| [7:5]       | Channel configure | For manual mode only: 12 bit ADC input channel select control bits.<br>adc12_ctrl<3:1> Selected channel<br>000 PGA inputs, differential<br>001 Temperature sensing inputs, differential<br>010 input A, positive and negative<br>011 input B, positive and negative<br>100 input C, positive and negative |
| [3]         | ADC enable        | 12b ADC power up control.<br>1: Power up ADC<br>0: Power down ADC   |

| Memory start/end addresses | ADC channels                      |
|----------------------------|-----------------------------------|
| 4005_0400 – 4005_047F      | PAG inputs, differential          |
| 4005_0480 – 4005_04FF      | Temperature sensing, differential |
| 4005_0500 – 4005_057F      | Input A, positive or differential |
| 4005_0580 – 4005_05FF      | Input A, negative                 |
| 4005_0600 – 4005_067F      | Input B, positive or differential |
| 4005_0680 – 4005_06FF      | Input B, negative                 |
| 4005_0700 – 4005_077F      | Input C, positive or differential |
| 4005_0780 – 4005_07FF      | Input C, negative                 |

| 0x4005_003C | ADC interrupt status | Register Description |
|-------------|----------------------|----------------------|
| [7]         |                      | input C, negative    |

|     |  |                                   |
|-----|--|-----------------------------------|
| [6] |  | Input C, positive or differential |
| [5] |  | Input B, negative                 |
| [4] |  | Input B, positive or differential |
| [3] |  | Input A, negative                 |
| [2] |  | Input A, positive or differential |
| [1] |  | Temperature sensing, differential |
| [0] |  | PGA inputs, differential          |

| 0x4005_0038 | ADC interrupt write clear | Register Description                                |
|-------------|---------------------------|---|
| [7]         |                           | input C, negative, write 1 to clear                 |
| [6]         |                           | Input C, positive or differential, write 1 to clear |
| [5]         |                           | Input B, negative, write 1 to clear                 |
| [4]         |                           | Input B, positive or differential, write 1 to clear |
| [3]         |                           | Input A, negative, write 1 to clear                 |
| [2]         |                           | Input A, positive or differential, write 1 to clear |
| [1]         |                           | Temperature sensing, differential, write 1 to clear |
| [0]         |                           | PGA inputs, differential, write 1 to clear          |

ADC can also be configured into auto channel sweep mode by setting the “adc\_ctrl\_override” bit to 0, with which the enabled channels will be sampled in the configured order automatically. The ten ADC input channels can be configured by programming their corresponding registers. Their configurations include sampling time, enable/disable, differential/single-ended, and continuous sampling/single-shot, based on the following register table. The sampled data is stored in the corresponding memory locations as in manual mode.

| 0x4000_F06C | ADC_CTL0                                     | Register Description  |
|-------------|--|---|
| [31:16]     | Temperature sensing, auto mode, differential | channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only |
| [15:0]      | PGA inputs, differential                     | channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only |

| 0x4000_F070 | ADC_CTL1           | Register Description  |
|-------------|--------------------|---|
| [31:16]     | Inputs A, negative | channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only |



|        |                                   |   |
|--------|-----------------------------------|---|
| [15:0] | Input A, positive or differential | channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only |
|--------|-----------------------------------|---|

| 0x4000_F074 | ADC_CTL2                          | Register Description  |
|-------------|-----------------------------------|---|
| [31:16]     | Input B, negative                 | channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only |
| [15:0]      | Input B, positive or differential | channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only |

| 0x4000_F078 | ADC_CTL3                          | Register Description  |
|-------------|-----------------------------------|---|
| [31:16]     | Input C, negative                 | channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only |
| [15:0]      | Input C, positive or differential | channel config: [3:0] sample time, for max rate 320k: 2T to 62T, step 4T; for max rate 256k, 3T to 63T, step 4T, T is period of 1.28MHz; [4] channel enable; [5] differential 1 or single-ended 0; [6] continuous 0 or one shot 1. For auto channel sweep mode only |

Table 13: ADC channel configurations

#### 4.14.3 ADC Channel <3:0> Connectivity

|              |                  |
|--------------|------------------|
| PGA inputs   | hardwired        |
| temp sensing | hardwired        |
| aio<0>       | Input A positive |
| aio<1>       | Input A negative |
| aio<2>       | Input B positive |
| aio<3>       | Input B negative |
| aio<4>       | Input C positive |
| aio<9>       | Input C negative |

**Table 15: ADC channel connectivity**

Aio<9, 4:0> and PGA inputs(Aio<7:8>) can be selected through an analog Mux by programming aio\_pass<7:0> or aio\_attn<7:0>. For example, register 0x4000\_F020<8><0> set to 01, then Aio<0> is connected to ADC input A positive node.

| 0x4000_F020 |                  | Register Description   |
|-------------|------------------|--|
| [13:8]      | Attenuation ctrl | attn[5:0]. analogIO control for {aio<9>, aio<4>, aio<3>, aio<2>, aio<1>, aio<0>}.<br>{attn[x], pass[x]}: |
|             |                  | 00 switch off  |
|             |                  | 01 pass  |
|             |                  | 10 attenuate to 1/4  |
|             |                  | 11 NC  |
| [5:0]       | pass ctrl        | pass[5:0]. analogIO control for {aio<9>, aio<4>, aio<3>, aio<2>, aio<1>, aio<0>}.<br>{attn[x], pass[x]}: |
|             |                  | 00 switch off  |
|             |                  | 01 pass  |
|             |                  | 10 attenuate to 1/4  |
|             |                  | 11 NC  |
|             |                  | note: analog IO sharing  |
|             |                  | gpio<11>/aio<0>  |
|             |                  | gpio<12>/aio<1>  |
|             |                  | gpio<13>/aio<2>  |
|             |                  | gpio<14>/aio<3>  |
|             |                  | gpio<15>/aio<4>  |
|             |                  | gpio<16>/aio<5>/32K XTAL input   |
|             |                  | gpio<17>/aio<6>/32K XTAL output  |
|             |                  | gpio<18>/aio<7>/pga in+  |
|             |                  | gpio<19>/aio<8>/pga in-  |
|             |                  | gpio<20>/aio<9>/mic bias   |

## 5 Absolute Maximum Ratings

Maximum ratings are the extreme limits to which CMT4501 can be exposed without permanently damaging it. Exposure to absolute maximum ratings for prolonged periods of time may affect the reliability of the CMT4501. **Table 16** specifies the absolute maximum ratings for CMT4501.

| Symbol                 | Parameter | Min. | Max. | Unit |
|------------------------|-----------|------|------|------|
| <b>Supply voltages</b> |           |      |      |      |
| VDD3                   |           | -0.3 | +3.6 | V    |
| DEC                    |           |      | 1.32 | V    |

|  |  |      |                   |                    |
|--|--|------|-------------------|--------------------|
| VSS  |  |      | 0                 | V                  |
| <b>I/O pin voltage</b>   |  |      |                   |                    |
| VIO  |  | -0.3 | VDD + 0.3         | V                  |
| <b>Environmental</b>   |  |      |                   |                    |
| Storage temperature  |  | -40  | +125              | °C                 |
| MSL  | Moisture Sensitivity Level                   |      | 3                 |                    |
| ESD HBM  | Human Body Model Class 2                     |      | 2                 | kV                 |
| ESD CDMQF  | Charged Device Model (QFN48, 7x7 mm package) |      | 500               | V                  |
| <b>Flash memory</b>  |  |      |                   |                    |
| Endurance  |  |      | 20 000            | write/erase cycles |
| Retention  |  |      | 10 years at 40 °C |                    |
| Number of times an address can be written between erase cycles |  |      |                   |                    |
|  |  |      | 2                 | times              |

Table 16: Absolute maximum ratings



## 6 Operating Conditions

The operating conditions are the physical Parameters that CMT4501 can operate within as defined in Table 17.

| Symbol | Parameter                       | Min. | Typ. | Max. | Units |
|--------|---------------------------------|------|------|------|-------|
| VDD3   | Supply voltage, normal mode     | 1.8  | 3    | 3.6  | V     |
| tr_VDD | Supply rise time (0 V to 1.8 V) |      |      | 100  | ms    |
| TA     | Operating temperature           | -40  | 27   | 125  | °C    |

Table 17: Operating conditions

## 7 Radio Transceiver

### 7.1 Radio Current Consumption

| Parameter       | Description             | MIN | TYP | MAX | UNIT |
|-----------------|-------------------------|-----|-----|-----|------|
| Tx only at 0dBm | with internal DC-DC @3V |     | 8   |     | mA   |
| Rx Only         | with internal DC-DC @3V |     | 8   |     | mA   |

Table 18: Radio current consumption

### 7.2 Transmitter Specification

| Parameter            | Description                                     | MIN | TYP  | MAX | UNIT |
|----------------------|---|-----|------|-----|------|
| RF Max Output Power  |   |     | 10   |     | dBm  |
| RF Min Output Power  |   |     | -20  |     | dBm  |
| OBW for BLE 1Mbps    | 20dB occupy-bandwidth for BLE modulation 1Mbps  |     | 1100 |     | KHz  |
| OBW for BLE 2Mbps    | 20dB occupy-bandwidth for BLE modulation 2Mbps  |     | 2300 |     | KHz  |
| OBW for GFSK 500Kbps | 20dB occupy-bandwidth for GFSK modulation 2Mbps |     | 1100 |     | KHz  |
| OBW for GFSK 125bps  | 20dB occupy-bandwidth for GFSK modulation 2Mbps |     | 1100 |     | KHz  |
| Error Vector Measure | Offset EVM for OQPSK modulation                 |     | 0.02 |     |      |
| FDEV for BLE 1Mbps   | Frequency deviation for GFSK modulation 1Mbps   | 160 |      | 250 | KHz  |
| FDEV for BLE 2Mbps   | Frequency deviation for GFSK modulation 2Mbps   | 320 |      | 500 | KHz  |

Table 19: Transmitter specification

### 7.3 Receiver Specification

### 7.3.1 RX BLE 1Mbps GFSK

| Parameter                          | Description   | MIN | TYP     | MAX | UNIT   |
|------------------------------------|---|-----|---------|-----|--------|
| Rx Sensitivity                     | Sensitivity test 1Mbps BLE ideal transmitter, 37 Byte BER=1E-3  |     | -97     |     | dBm    |
| co-channel rejection               | modulated interferer in channel, 37 Byte BER=1E-3   |     | -6      |     | I/C dB |
| Selectivity +/-1MHz                | Wanted signal at -67dBm, modulated interferer at +/- 1MHz, 37 Byte BER=1E-3   |     | 7       |     | I/C dB |
| Selectivity +/-2MHz                | Wanted signal at -67dBm, modulated interferer at +/- 2MHz, 37 Byte BER=1E-3   |     | 45      |     | I/C dB |
| Selectivity +/-3MHz                | Wanted signal at -67dBm, modulated interferer at +/- 3MHz, 37 Byte BER=1E-3   |     | 50      |     | I/C dB |
| Selectivity +/-4MHz                | Wanted signal at -67dBm, modulated interferer at +/- 4MHz, 37 Byte BER=1E-3   |     | 50      |     | I/C dB |
| Selectivity +/-5MHz or More        | Wanted signal at -67dBm, modulated interferer at >= +/- 5MHz, 37 Byte BER=1E-3  |     | 55      |     | I/C dB |
| Selectivity Imag frequency         | Wanted signal at -67dBm, modulated interferer at imagefrequency, 37 Byte BER=1E-3   |     | 22      |     | I/C dB |
| Intermodulation                    | Wanted signal at 2402MHz, -64dBm, Two interferers at 2405 and 2408 MHz respectively, at the given power level, 37 Byte BER=1E-3 |     | -20     |     | dBm    |
| Carrier Frequency Offset Tolerance |   |     | +/- 350 |     | KHz    |
| Sample Clock Offset Tolerance      |   |     | +/- 120 |     | ppm    |

Table 20: RX BLE 1Mbps GFSK

### 7.3.2 RX BLE 2Mbps GFSK

| Parameter                   | Description   | MIN | TYP | MAX | UNIT   |
|-----------------------------|---|-----|-----|-----|--------|
| Rx Sensitivity              | Sensitivity test 2Mbps BLE ideal transmitter, 37 Byte BER=1E-3                    |     | -94 |     | dBm    |
| co-channel rejection        | modulated interferer in channel, 37 Byte BER=1E-3                                 |     | -6  |     | I/C dB |
| Selectivity +/-1MHz         | Wanted signal at -67dBm, modulated interferer at +/- 1MHz, 37 Byte BER=1E-3       |     | -5  |     | I/C dB |
| Selectivity +/-2MHz         | Wanted signal at -67dBm, modulated interferer at +/- 2MHz, 37 Byte BER=1E-3       |     | 9   |     | I/C dB |
| Selectivity +/-3MHz         | Wanted signal at -67dBm, modulated interferer at +/- 3MHz, 37 Byte BER=1E-3       |     | 30  |     | I/C dB |
| Selectivity +/-4MHz         | Wanted signal at -67dBm, modulated interferer at +/- 4MHz, 37 Byte BER=1E-3       |     | 40  |     | I/C dB |
| Selectivity +/-5MHz or More | Wanted signal at -67dBm, modulated interferer at >= +/- 5MHz, 37 Byte BER=1E-3    |     | 55  |     | I/C dB |
| Selectivity Imag frequency  | Wanted signal at -67dBm, modulated interferer at imagefrequency, 37 Byte BER=1E-3 |     | 22  |     | I/C dB |

|                                    |   |  |         |  |     |
|------------------------------------|---|--|---------|--|-----|
| Intermodulation                    | Wanted signal at 2402MHz, -64dBm, Two interferers at 2405 and 2408 MHz respectively, at the given power level, 37 Byte BER=1E-3 |  | -20     |  | dBm |
| Carrier Frequency Offset Tolerance |   |  | +/- 350 |  | KHz |
| Sample Clock Offset Tolerance      |   |  | +/- 120 |  | ppm |

Table 21: RX BLE 2Mbps GFSK

### 7.3.3 RX 500Kbps GFSK

| Parameter                          | Description   | MIN | TYP    | MAX | UNIT   |
|------------------------------------|---|-----|--------|-----|--------|
| Rx Sensitivity                     | Sensitivity test 500Kbps BLE ideal transmitter, 37 Byte BER=1E-3  |     | -98    |     | dBm    |
| co-channel rejection               | modulated interferer in channel, 37 Byte BER=1E-3   |     | -4     |     | I/C dB |
| Selectivity +/- 1MHz               | Wanted signal at -67dBm, modulated interferer at +/- 1MHz, 37 Byte BER=1E-3   |     | 10     |     | I/C dB |
| Selectivity +/- 2MHz               | Wanted signal at -67dBm, modulated interferer at +/- 2MHz, 37 Byte BER=1E-3   |     | 45     |     | I/C dB |
| Selectivity +/- 3MHz               | Wanted signal at -67dBm, modulated interferer at +/- 3MHz, 37 Byte BER=1E-3   |     | 50     |     | I/C dB |
| Selectivity +/- 4MHz               | Wanted signal at -67dBm, modulated interferer at +/- 4MHz, 37 Byte BER=1E-3   |     | 50     |     | I/C dB |
| Selectivity +/- 5MHz or More       | Wanted signal at -67dBm, modulated interferer at >= +/- 5MHz, 37 Byte BER=1E-3  |     | 55     |     | I/C dB |
| Selectivity Imag frequency         | Wanted signal at -67dBm, modulated interferer at imagefrequency, 37 Byte BER=1E-3   |     | 24     |     | I/C dB |
| Intermodulation                    | Wanted signal at 2402MHz, -64dBm, Two interferers at 2405 and 2408 MHz respectively, at the given power level, 37 Byte Ber=1E-3 |     | -19    |     | dBm    |
| Carrier Frequency Offset Tolerance |   |     | +/-350 |     | KHz    |
| Sample Clock Offset Tolerance      |   |     | +/-120 |     | ppm    |

Table 22: RX 500Kbps GFSK

### 7.3.4 RX 125Kbps GFSK

| Parameter            | Description   | MIN | TYP  | MAX | UNIT   |
|----------------------|---|-----|------|-----|--------|
| Rx Sensitivity       | Sensitivity test 125Kbps BLE ideal transmitter, 37 Byte BER=1E-3            |     | -103 |     | dBm    |
| co-channel rejection | modulated interferer in channel, 37 Byte BER=1E-3                           |     | -1   |     | I/C dB |
| Selectivity +/- 1MHz | Wanted signal at -67dBm, modulated interferer at +/- 1MHz, 37 Byte BER=1E-3 |     | -11  |     | I/C dB |
| Selectivity +/- 2MHz | Wanted signal at -67dBm, modulated interferer at +/- 2MHz, 37 Byte BER=1E-3 |     | 45   |     | I/C dB |
| Selectivity +/- 3MHz | Wanted signal at -67dBm, modulated interferer at +/- 3MHz, 37 Byte BER=1E-3 |     | 50   |     | I/C dB |

|                                    |   |  |         |  |        |
|------------------------------------|---|--|---------|--|--------|
| Selectivity +/- 4MHz               | Wanted signal at -67dBm, modulated interferer at +/- 4MHz, 37 Byte BER=1E-3   |  | 50      |  | I/C dB |
| Selectivity +/- 5MHz or More       | Wanted signal at -67dBm, modulated interferer at >= +/- 5MHz, 37 Byte BER=1E-3  |  | 55      |  | I/C dB |
| Selectivity Imag frequency         | Wanted signal at -67dBm, modulated interferer at imagefrequency, 37 Byte BER=1E-3   |  | 28      |  | I/C dB |
| Intermodulation                    | Wanted signal at 2402MHz, -64dBm, Two interferers at 2405 and 2408 MHz respectively, at the given power level, 37 Byte BER=1E-3 |  | -18     |  | dBm    |
| Carrier Frequency Offset Tolerance |   |  | +/- 350 |  | KHz    |
| Sample Clock Offset Tolerance      |   |  | +/- 120 |  | ppm    |

Table 23: RX 125Kbps GFSK

## 7.4 RSSI Specifications

| Parameter          | Description                                 | MIN | TYP  | MAX | UNIT |
|--------------------|---|-----|------|-----|------|
| RSSI Dynamic Range |   |     | 70   |     | dB   |
| RSSI Accuracy      | RSSI Accuracy Valid in range -100 to -30dBm |     | +/-2 |     | dB   |
| RSSI Resolution    | Totally 7bit, from 0 to 127                 |     | 1    |     | dB   |
| RSSI Period        |   |     | 8    |     | us   |

Table 24: RSSI specifications

## 8 Glossary

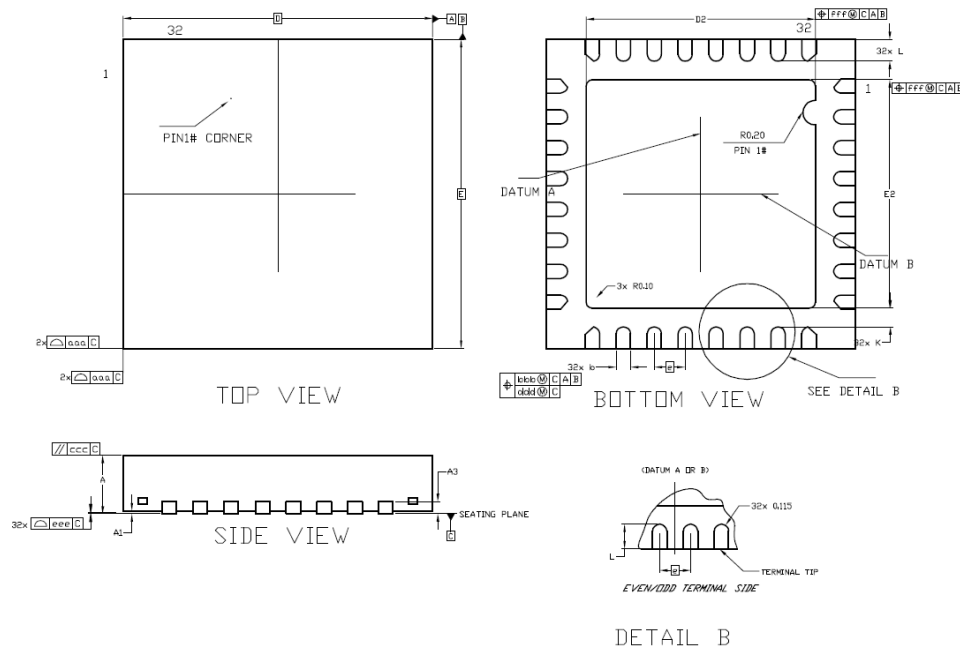
| Term    | Description  |
|---------|--|
| AHB     | Advanced High-performance Bus (ARM bus standard)     |
| AHB-AP  | DAP AHB Port for debug component access thru AHB bus |
| AMBA    | Advanced Microcontroller Bus Architecture            |
| AON     | Always-on power domain                               |
| APB     | Advanced Peripheral Bus (ARM bus standard)           |
| APB-AP  | DAP APB Port for debug component access thru APB bus |
| BROM    | Boot ROM   |
| DAP     | Debug Access Port ( ARM bus standard)                |
| ETM     | Embedded trace module                                |
| FPU     | Floating Point Unit                                  |
| I2C     | Inter-Integrated Circuit                             |
| I2S     | Inter-IC Sound, Integrated Interchip Sound           |
| ITM     | Instrumentation Trace Macrocell Unit                 |
| JTAG    | Joint Test Access Group (IEEE standard)              |
| JTAG-AP | DAP's JTAG Access Port to access debug components    |
| JTAG-DP | DAP's JTAG Debug Port used by external debugger      |
| J&M     | Jun and Marty LLC                                    |
| MPU     | Memory Protection Unit                               |
| NVIC    | Nested vector Interrupt Controller                   |
| PCR     | Power Clock Reset controller                         |
| POR     | Power on reset, it is active low in this document    |
| RFIF    | APB peripheral to interface RF block                 |
| SWD     | Serial Wire DAP (ARM bus standard)                   |
| SoC     | System on chip                                       |
| SPI     | Serial Peripheral Interface                          |
| SRAM    | Static Random Access memory                          |
| TWI     | Two-Wire Interface                                   |
| UART    | Universal Asynchronous Receiver and Transmitter      |
| WDT     | Watchdog Timer                                       |

**Table 25: Glossary**



## 9 Package dimensions

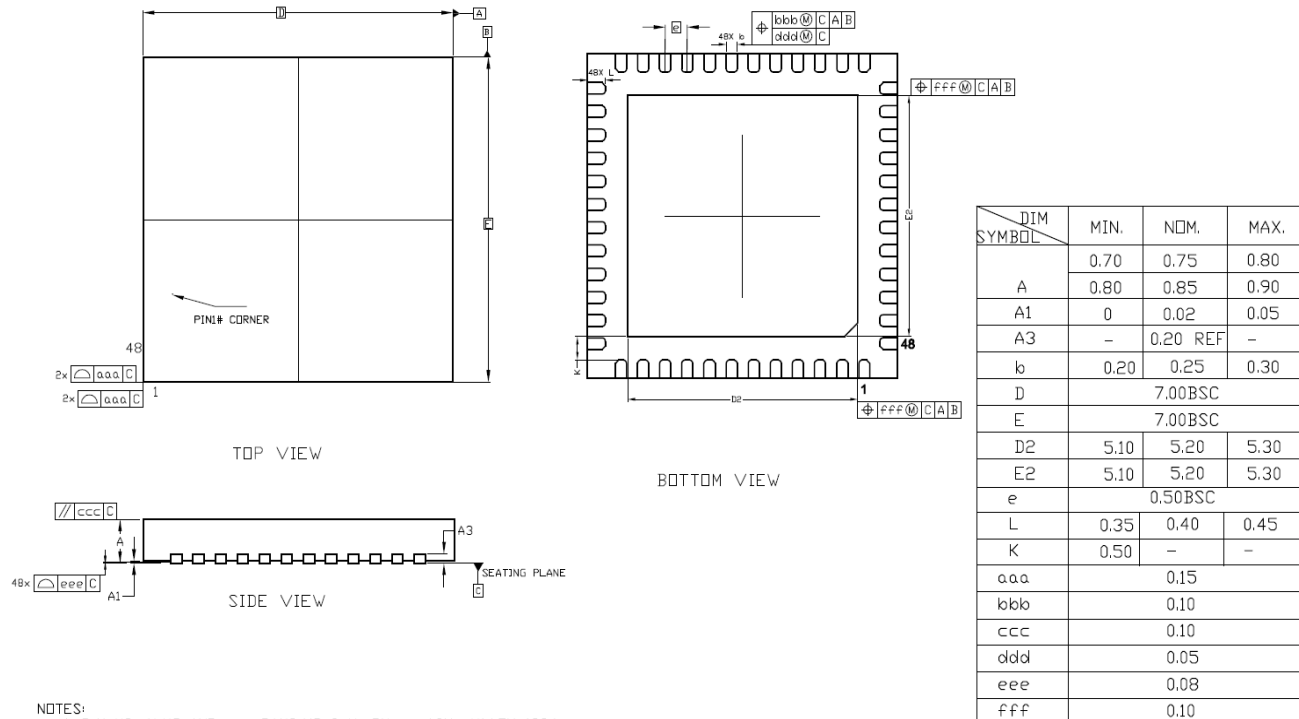
### QFN32 package dimensions:



NOTES:  
1. DIMENSIONING AND TOLERANCING CONFIRM TO ASME Y14.5M-1994

**Note: dimensions are in mm, angels are in degree.**

### QFN48 package dimensions:

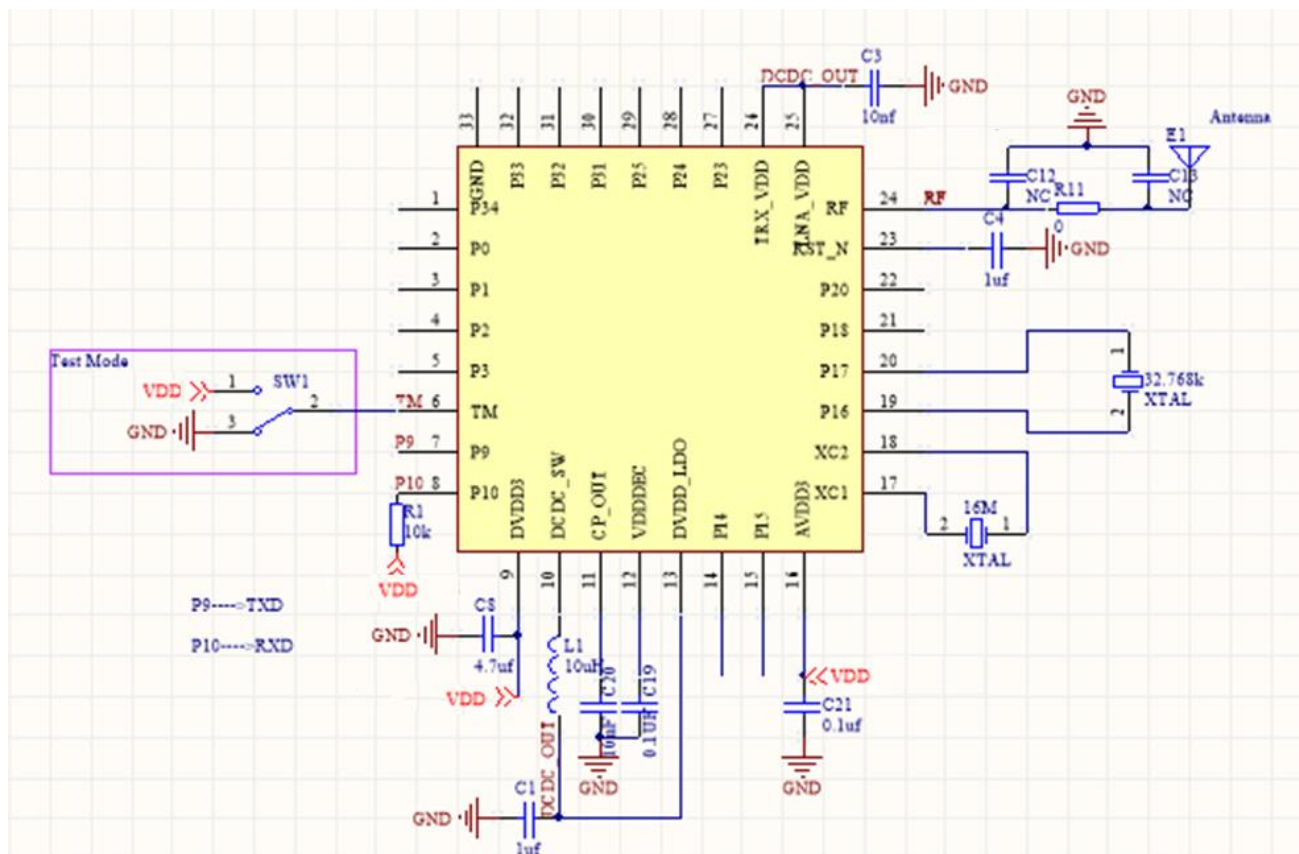


NOTES:  
1. DIMENSIONING AND TOLERANCING CONFIRM TO ASME Y14.5M-1994

**Note: dimensions are in mm, angels are in degree.**

## 10 Sample Application and Layout Guide

### 1) Sample Application



### 2) Layout Guide

#### Placement

1. RF matching/Loop filter leading to antenna should be isolated from any other AC/DC signal as much as possible;
2. Xtal/OSC clock is a noise source to other circuits, keep clock trace as short as possible and away from any important area;
3. LDO's are sensitive and could be easily contaminated, care should be taken for the environment;
4. Antenna is the main RF radiation point, other important blocks should be shielded or away from this area.

#### RF traces

1. Define RF line width with given dielectric thickness (thickness of PCB dielectric layer to ground plane) to achieve 50ohm impedance; this is mainly for the RF line connecting to matching/loop filter and antenna.
2. Differential traces should be kept in the same length and component should be placed symmetrically;
3. Certain length of RF trace should be treated as part of RF matching.

#### Bypass Capacitor

1. Each VDD pin needs a bypass capacitor to release chip internal noise and block noise from power supply.
2. For power traces, bypass capacitors should be placed as close as possible to VDD pins.

3. Use one large and one small capacitor when the pin needs two capacitors. Typically the capacitance of the larger capacitor is about 100 times of that of the smaller one. The smaller capacitor usually has better quality factor than the larger one. Place the larger capacitor closer to the pin.
4. The capacitors of Loop filter need to have larger clearance to prevent EMC/EMI issue.
5. Ground via should be close to the Capacitor GND side, and away from strong signals.

#### Layer Definition

1. Normally 4 layer PCB is recommended.
2. RF trace must be on the surface layer, i.e. top layer or bottom.
3. The second layer of RF PCB must be "Ground " layer , for both signal ground and RF reference ground , DO NOT put any other trace or plane on second layer, otherwise "antenna effect" will complicate debug process.
4. Power plane generally is on the 3rd layer.
5. Bottom layer is for "signal " layer.
6. If two layer PCB is used, quality will degrade in general. More care needs to be taken. Try to maximize ground plane, avoid crossing of signal trace with other noise lines or VDD, shield critical signal line with ground plane, maximize bypass capacitor and number of ground vias.

#### Reference clock and trace

1. Oscillator signal trace is recommended to be on the 1<sup>st</sup> layer;
2. DO NOT have any trace around or across the reference clock (oscillator) trace.
3. Isolate the reference clock trace and oscillator by having more GND via around.
4. DO NOT have any other traces under the Oscillator.

#### Power line or plane

1. Whether to use power plain or power line depend on the required current, noise and layout condition. For RF chip, we generally suggest to use power line to bring power into IC pin. Line has parasitic inductance, which forms a low pass filter to reduce the noise traveling around PCB.
2. Add more conductive via on the current source, it will increase max current limit and reduce inductance of via.
3. Add some capacitor alone the power trace when power line travels a long distance.
4. DO NOT place power line or any plane under RF trace or oscillator and its clock trace , the strong clock or RF signal would travel with power line.

#### Ground Via

1. Ground Via must be as close to the ground pad of bypass capacitor as possible , too much distance between via and ground pad will reduce the effect of bypass capacitor.
2. Having as many ground via as possible.
3. Place ground via around RF trace, the RF trace should be shielded with via trail.

## **Liability Disclaimer**

CMOSTEK Inc. reserves the right to make changes without further notice to the product to improve reliability, function or design. CMOSTEK Inc. does not assume any liability arising out of the application or use of any product or circuits described herein.

## **Life Support Applications**

CMOSTEK Semiconductor's products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. CMOSTEK Inc. customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify CMOSTEK Inc. for any damages resulting from such improper use or sale.

## **Contacts**

CMOSTEK Microelectronics Co., Ltd. Shenzhen Branch

Address: 2/F Building 3, Pingshan Private Enterprise S.T. Park, Xili, Nanshan District, Shenzhen, Guangdong, China

**Tel:** +86-755-83231427

**Post Code:** 518057

**Sales:** [sales@cmostek.com](mailto:sales@cmostek.com)

**Supports:** [support@cmostek.com](mailto:support@cmostek.com)

**Website:** [www.cmostek.com](http://www.cmostek.com)